

CR-19 3982



12214 Lakewood Blvd., Downey, California 90241, (310) 922-3803

N95-10870

Unc l

63/34 0019680

**ZERO-G THERMODYNAMIC VENTING SYSTEM (TVS)
PERFORMANCE PREDICTION PROGRAM**

PREPARED BY

Han Nguyen
**HAN NGUYEN
MPS ANALYSIS UNIT
CRYOGENIC SYSTEMS GROUP
VEHICLE AND SYSTEMS DESIGN**

(NASA-CR-193982) ZERO-G
THERMODYNAMIC VENTING SYSTEM (TVS)
PERFORMANCE PREDICTION PROGRAM
Final Report (Rockwell
International Corp.) 144 p

APPROVED BY

Tibor Lak
**TIBOR LAK
TECHNICAL MANAGER**

Charles C. Wood
**CHARLES C. WOOD
PROGRAM MANAGER**

No. of Pages: 138

Date: May 24, 1994

100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100

100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100 100



12214 Lakewood Blvd., Downey, California 90241, (310) 922-3803

**ZERO-G THERMODYNAMIC VENTING SYSTEM (TVS)
PERFORMANCE PREDICTION PROGRAM**

PREPARED BY

Han Nguyen

**HAN NGUYEN
MPS ANALYSIS UNIT
CRYOGENIC SYSTEMS GROUP
VEHICLE AND SYSTEMS DESIGN**

APPROVED BY

Tibor Lak

**TIBOR LAK
TECHNICAL MANAGER**

Charles C. Wood

**CHARLES C. WOOD
PROGRAM MANAGER**

ABSTRACT

This report documents the Zero-g Thermodynamic Venting System (TVS) performance prediction computer program. The zero-g TVS is a device that destratifies and rejects environmentally induced zero-g thermal gradients in the LH₂ storage transfer system. A recirculation pump and spray injection manifold recirculates liquid throughout the length of the tank, thereby destratifying both the ullage gas and liquid bulk. Heat rejection is accomplished by the opening of the TVS control valve which allows a small flow rate to expand to a low pressure thereby producing a low temperature heat sink which is used to absorb heat from the recirculating liquid flow. The program was written in FORTRAN 77 language on the HP-9000 and IBM PC computers. It can be run on various platforms with a FORTRAN compiler.

TABLE OF CONTENTS

<u>SECTION</u>	<u>PAGE</u>
Title Page	i
Abstract	ii
Table of Contents	iii
List of Illustrations	vi
1 INTRODUCTION	I-1
1.1 Purpose	I-1
1.2 Problem Description	I-1
1.3 Areas of Applications	I-1
1.4 Description of the Physical Model	I-1
2 ANALYTICAL MODEL DESCRIPTION	II-1
2.1 Heat Exchanger Model	II-1
2.1.1 Fluid Quality at Heat Exchanger Outlet	II-1
2.1.2 Two-Phase Pressure Loss in the Heat Exchanger	II-2
2.1.3 Forced Convection Heat-Transfer Model	II-3
2.1.4 Spray Temperature	II-4
2.2 Spray Manifold and Injection Tube Model	II-4
2.2.1 Spray Manifold	II-7
2.2.2 Spray Injection Tube	II-9
2.2.3 Spray Manifold and Injection Tube Model Algorithm	II-12
2.3 Recirculation Pump Model	II-12
2.4 Tank Thermal Model	II-15
2.4.1 Ullage	II-15
2.4.2 Tank Wall	II-19
2.4.3 Wall Liquid	II-19
2.4.4 Bulk Liquid	II-21
2.4.5 Heat Transfer	II-24
2.4.5.1 Free Convection	II-25
2.4.5.2 Forced Convection	II-25

2.4.6 Mass Transfer	II-26
2.4.6.1 Bulk Liquid Boiling	II-26
2.4.6.2 Wall Liquid Boiling	II-27
2.4.6.3 Liquid Droplet Evaporation in the Ullage	II-27
2.4.6.4 Liquid Spray Falling into Bulk Liquid or Accumulating on Tank Wall	II-28
2.4.6.5 Ullage Condensation	II-28
2.4.6.6 Liquid Surface Condensation	II-28
2.5 References	II-28
2.6 Sample Cases	II-31
3.0 COMPUTER MODEL DESCRIPTION	III-1
3.1 Programming Description	III-1
3.2 Flow Charts	III-1
3.2.1 Heat Exchanger Model	III-1
3.2.2 Spray Manifold/Injection Tube Model	III-2
3.2.3 Integrated Zero-g TVS Model	III-3
3.3 Description of Variables	III-5
3.3.1 Input Variables	III-5
3.3.1.1 Heat Exchanger Model	III-5
3.3.1.2 Spray Manifold/Injection Tube Model	III-5
3.3.1.3 Recirculation Pump Model	III-5
3.3.1.4 Integrated Zero-g TVS Model	III-6
3.3.1.5 LH ₂ Saturated Properties	III-8
3.3.1.6 GH ₂ Properties	III-8
3.3.2 Output Variables	III-9
3.3.2.1 Heat Exchanger Model	III-9
3.3.2.2 Spray Manifold/Injection Tube Model	III-9
3.3.2.3 Integrated Zero-g TVS Model	III-10
3.4 Program Listing	III-12
3.4.1 Heat Exchanger Model	III-12

3.4.2 Integrated Zero-g TVS Model	III-56
3.5 Input Data	III-85
 3.5.1 Heat Exchanger Model	III-85
 3.5.2 Integrated Zero-g TVS Model	III-95

LIST OF ILLUSTRATIONS

FIGURE	TITLE	PAGE
1.1	Zero-g Thermodynamic Venting System (TVS) Concept	I-2
2.2.1	Spray Manifold Model	II-8
2.2.2	Spray Injection Tube Model	II-11
2.3.1	LH ₂ Recirculation Pump Head-Flow Curve	II-14
2.4.1	Tank Thermal Model	II-16
2.4.2	Ullage Thermal Model	II-17
2.4.3	Tank Wall Thermal Model	II-20
2.4.4	Wall Liquid Thermal Model	II-20
2.4.5	Bulk Liquid Thermal Model	II-22
2.4.6	Free Convection Heat-Transfer Correlation for Interior Surfaces of Vertical Ducts, Vertical Plates and Cylinders, and Horizontal Cylinders	II-26
2.4.7	Forced Convection Heat-Transfer Correlation for Flow over a Sphere	II-27
2.4.8	Droplet Evaporation Model	II-31
2.4.9	Ullage Condensation Model	II-31
2.6.1	TVS Performance Simulation at 90% Liquid Quantity	II-34
2.6.2	TVS Recirculation Pump and Vent Valve Flow Rate Transient During Ullage Destratification(90% Liquid Quantity)	II-34
2.6.3	TVS Performance Simulation at 25% Liquid Quantity	II-35
2.6.4	TVS Performance Simulation at 25% Liquid Quantity	II-35
2.6.5	TVS Recirculation Pump and Vent Valve Flow Rate Transient During Ullage Destratification (25% Liquid Quantity)	II-36
2.6.6	TVS Operation Frequency (Percent) as a Function of Liquid Quantity	II-36

SECTION 1

INTRODUCTION

1.1 Purpose

The purpose of the zero-g thermodynamic venting system (TVS) model is to define the pressure level requirements, propellant loss due to venting, total pump power consumption, venting system operation duration and frequency as a function of liquid level and acceleration environment.

1.2 Problem Description

Long-term storage of subcritical cryogens in space is subjected to thermal stratification which is more severe than experienced in a 1-g environment due to the absence of gravity-induced body forces. If left uncontrolled, the thermal gradients result in excessive tank pressure rise and formation of liquid/vapor mixtures within the liquid bulk, liquid acquisition device, and propellant transfer lines. A subsystem is therefore needed to reduce the thermal gradient to acceptable levels, and reject the environmental heat leakage in an efficient manner.

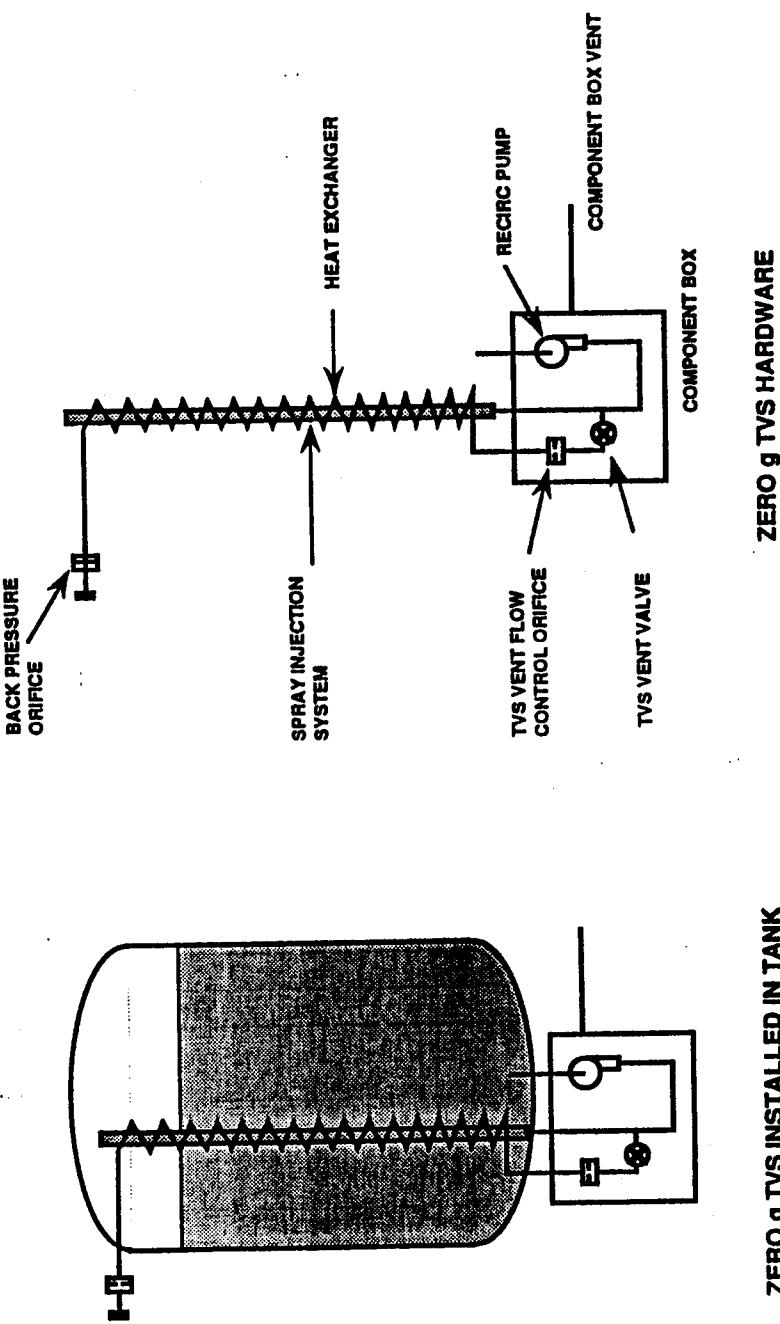
1.3 Areas of Application

This program is designed to predict the tank ullage pressure and temperature, pump flow rate and power consumption, propellant loss due to venting, venting duration and frequency for different liquid levels, accelerations and heat leak rates. It can also be used to model tank chill down and liquid fill in zero-g and one-g environments. The program therefore has general applicability in areas of long-term cryogenic fluid storage and transfer in space and on the ground.

1.4 Description of the Physical Model

The zero-g TVS concept, shown in Figure 1.1, has been defined to operate in a self-induced, forced convection environment. It includes a recirculation pump, located external to the tank, to flow liquid from the tank, a spray manifold and injection tube to mix and destratify the ullage and liquid bulk, a heat exchanger to absorb heat from the tank, and an overboard vent line to reject the heat. The concept operates in the following manner. When the vent pressure level is reached, the recirculation pump is activated, resulting in liquid flow through the spray injection manifold which destratifies the liquid bulk and ullage gas through forced convection. When the fluid bulk temperature reaches a predetermined level, the vent valve is opened, resulting in a temperature drop of the vent flow by isenthalpic expansion. This liquid is used to absorb heat from the recirculation flow via the heat exchanger, and subsequently vented overboard as a gas.

Figure 1.1 Zero-g Thermodynamic Venting System (TVS) Concept



SECTION 2

ANALYTICAL MODEL DESCRIPTION

The zero-g TVS model consists of the thermal-fluid models of the heat exchanger, spray manifold and injection tubes, recirculation pump, and tank. These models were developed and verified independently before they were integrated into the transient TVS model. Following is a description of each model.

2.1 Heat Exchanger Model

The heat exchanger model is based on the generalized two-phase cryogenic propellant dump model developed to evaluate the Space Shuttle Main Propulsion System (MPS) cryogenic propellant dump/vacuum inerting operations performance (Ref. 29). It is a multi-node finite difference model that simulates two-phase flow in a quasi steady-state mode.

The model uses the fluid properties at the inlet of the spray manifold as the input to the first node. With one of the inlet fluid property, namely the total enthalpy of the fluid at the inlet, the total enthalpy at the exit can be calculated based on the First Law of Thermodynamics. The total enthalpy at the exit, and an assumed mass flow rate are used to determine the exit static pressure. The exit static pressure is determined by an iterative process: with the flow assumed choked at the exit, the exit static pressure is increased incrementally until the maximum entropy is achieved (sonic flow), or when it becomes greater than the back pressure (subsonic flow). From the calculated exit pressure, the other exit fluid properties of the last node, the total pressure loss between the inlet and outlet can then be calculated and the inlet fluid properties can be determined.

The following sections will provide the equations used in the heat exchanger model.

2.1.1 Fluid Quality At Heat Exchanger Outlet

The outlet static pressure is calculated, assuming choked or sonic flow, where the entropy point is maximum. The following equations are solved simultaneously for the liquid quality of the fluid at the outlet

$$h_o = h_i + \frac{Q}{m} + \Delta Ha \quad (2.1.1)$$

$$V_o = \frac{m}{\rho_o A} \quad (2.1.2)$$

$$\rho_o = \frac{1}{\frac{1}{(\rho_L)_o} + Y_o \left[\frac{1}{(\rho_v)_o} - \frac{1}{(\rho_L)_o} \right]} \quad (2.1.3)$$

$$h_o = (1 - y_o)(h_L)_o + Y_o(h_v)_o + \frac{V_o^2}{2g_e J} \quad (2.1.4)$$

where the following fluid properties are based on the outlet static pressure

$(h_L)_o$ = the outlet liquid enthalpy

$(h_v)_o$ = the outlet vapor enthalpy

$(\rho_L)_o$ = the outlet liquid density

$(\rho_v)_o$ = the outlet vapor density

Q = total heat-transfer rate to a specific node

ΔH = change in height of the line between inlet and outlet

$a = \frac{g}{g_c}$ = acceleration

h_o = total enthalpy at the outlet

ρ_o = total density at the outlet

V_o = fluid velocity at the outlet

Y_o = fluid quality at the outlet

With the outlet quality, the total entropy then can be calculated using the following equation

$$\{(S)_o = (1 - y_o)(S_L)_o + Y_o(S_v)_o\}_{max} \quad (2.1.5)$$

where $(S_L)_o$ = the outlet liquid entropy

$(S_v)_o$ = the outlet vapor entropy

Iteration of the above outlet equations can be performed to obtain the maximum entropy point and the outlet static pressure.

2.1.2 Two-Phase Pressure Loss in the Heat Exchanger

To calculate the two-phase pressure loss (momentum and friction) between the inlet and outlet of the heat exchanger, the Lockhart-Martinelli correlation is used. The outlet pressure is

$$P_o = (P_s)_o + (P_d)_o \quad (2.1.6)$$

The total pressure loss term is further defined as

$$\Delta P_T = \Delta P_m + \Delta P_f \quad (2.1.7)$$

where ΔP_m = pressure loss due to momentum change

ΔP_f = pressure loss due to frictional forces

The momentum pressure loss is defined as

$$\Delta P_m = \frac{\dot{m}}{g_c A} (V_o - V_i) \quad (2.1.8)$$

The frictional pressure loss is defined as

$$\Delta P_f = \frac{144K}{2\rho_L g_c} \left[\frac{m(1-\bar{Y})}{A} \right]^2 \Phi_L^2 \quad (2.1.9)$$

where $K = \left(f \frac{L}{D} \right)$ = the line loss coefficient

$\bar{\rho}_L$ = the average liquid density between the inlet and outlet

\bar{Y} = the average total liquid quality between the inlet and outlet

$\Phi_L = f(X)$ is the Lockhart-Martinelli correlation factor

The Lockhart-Martinelli correlation is approximately defined as

$$\Phi_L^2 = 1 + \frac{1}{X} + \frac{1}{X^2} \quad (2.1.10)$$

X is defined as

$$X = \left(\frac{\bar{\mu}_L}{\bar{\mu}_v} \right)^{0.1} \left(\frac{1 - \bar{Y}}{\bar{Y}} \right)^{0.9} \left(\frac{\bar{\rho}_v}{\bar{\rho}_L} \right)^{0.5} \quad (2.1.11)$$

where $\bar{\rho}_v$ = the average vapor density between the inlet and outlet

$\bar{\mu}_L$ = the average liquid viscosity between the inlet and outlet

$\bar{\mu}_v$ = the average vapor viscosity between the inlet and outlet

2.1.3 Forced Convection Heat -Transfer Model

The heat-transfer equations used in the steady-state model are as follows

Two-phase heat transfer using the correlation proposed by John C. Chen (1963)

$$\frac{Q}{A} = [h_{FC}F + h_{FZ}S]\Delta T \quad (2.1.12)$$

where

$$h_{FC} = 0.023 \left(\frac{DG}{\mu_L} \right)^{0.8} \left(\frac{\mu_L C_L}{k_L} \right)^{0.4} \left(\frac{k_L}{D} \right) \quad (2.1.13)$$

$$h_{FZ} = 0.00122 \frac{k_L^{0.79} C_L^{0.45} \rho_L^{0.49} g_c^{0.25} \Delta T^{0.24} \Delta P^{0.75}}{\sigma^{0.5} \mu_L^{0.29} \lambda^{0.24} \rho_v^{0.24}} \quad (2.1.14)$$

$$F = f(X_a) \quad (2.1.15)$$

$$Re_L = \frac{DG(1-Y)}{\mu_L} \quad (2.1.16)$$

$$\Delta T = T_w - T_s \quad (2.1.17)$$

$$\Delta P = \frac{\Delta T \rho_v \lambda}{T_s} \quad (2.1.18)$$

The single-phase heat-transfer correlation used in the model (liquid and superheated gas)

$$\frac{Q}{A} = h \Delta T \quad (2.1.19)$$

$$h = 0.023 \left(\frac{DG}{\mu_L} \right)^{0.8} \left(\frac{\mu_L C_L}{k_L} \right)^{0.4} \left(\frac{k_L}{D} \right) \quad (2.1.20)$$

2.1.4 Spray Temperature

To shorten the run time of the zero-g TVS model, an alternative was provided to calculate the spray temperature as a function of the liquid subcooling and tank pressure. This is obtained from an energy balance between the hot and cold fluids of the heat exchanger

$$T_s = T_p - \frac{m_v \Delta h}{m_p c_{pl}} \quad (2.1.21)$$

where T_s is the spray temperature

T_p is the pump temperature

m_v is the vent flow rate

m_p is the pump flow rate

Δh is the heat absorption capability of the vent flow

c_{pl} is the heat capacity of the liquid

The TVS vent flow rate and heat absorption capability were calculated as a function of the liquid subcooling and tank flow rate and are shown in Figs. 2.1 and 2.2, respectively. This data is provided as a table look-up to the zero-g TVS model.

2.2 Spray Manifold and Injection Tube Model

Fluid is recirculated from the tank to the spray manifold and injection tubes where it is sprayed into the ullage and liquid. A one-dimensional, incompressible fluid dynamic model was developed to determine the pressures in the spray manifold and injection tubes, and to calculate the spray flow rates and velocities leaving the injection orifices. Following is a description of the model.

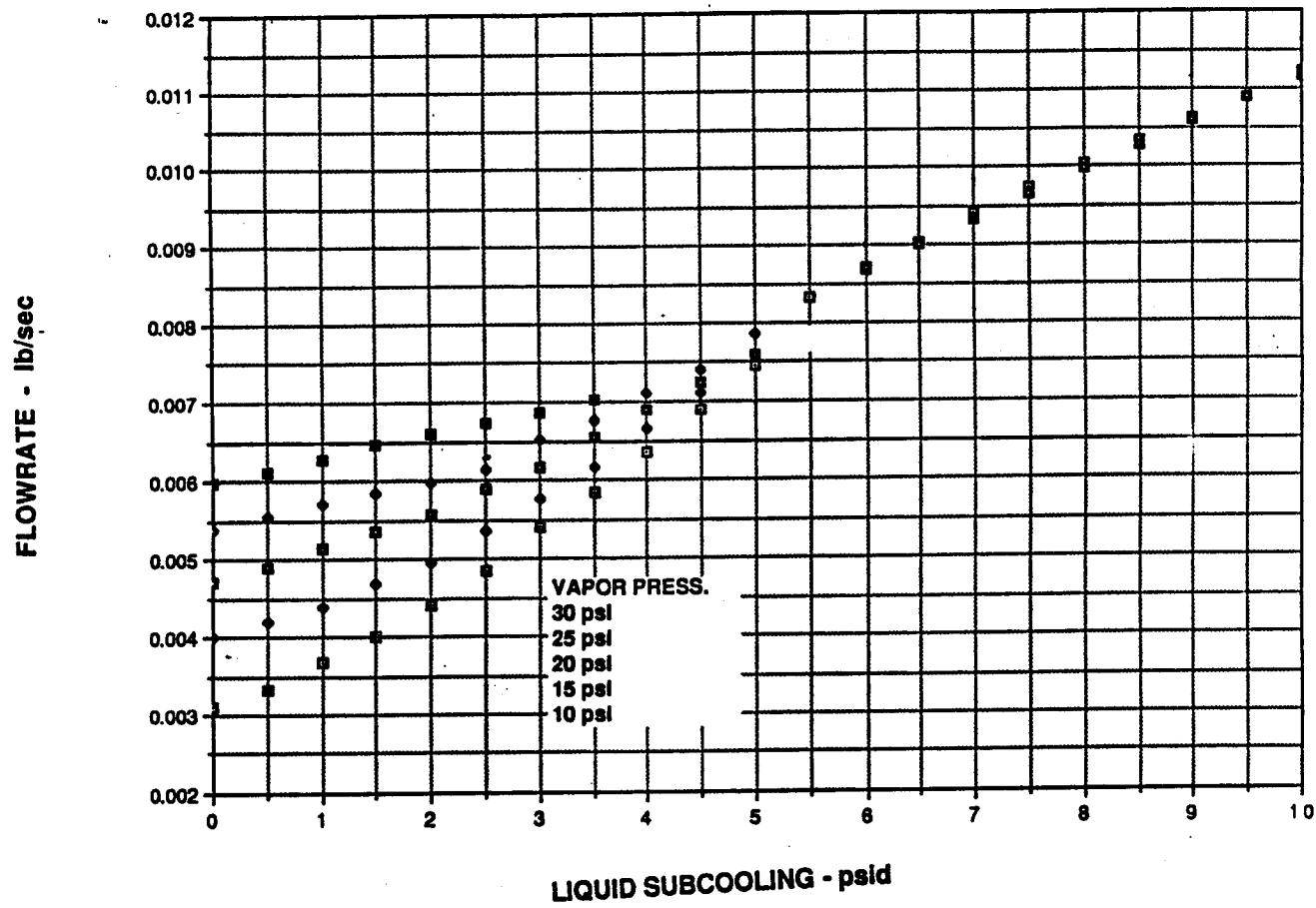


Figure 2.1.1 TVS Vent Flow Rate as a Function of Liquid Subcooling

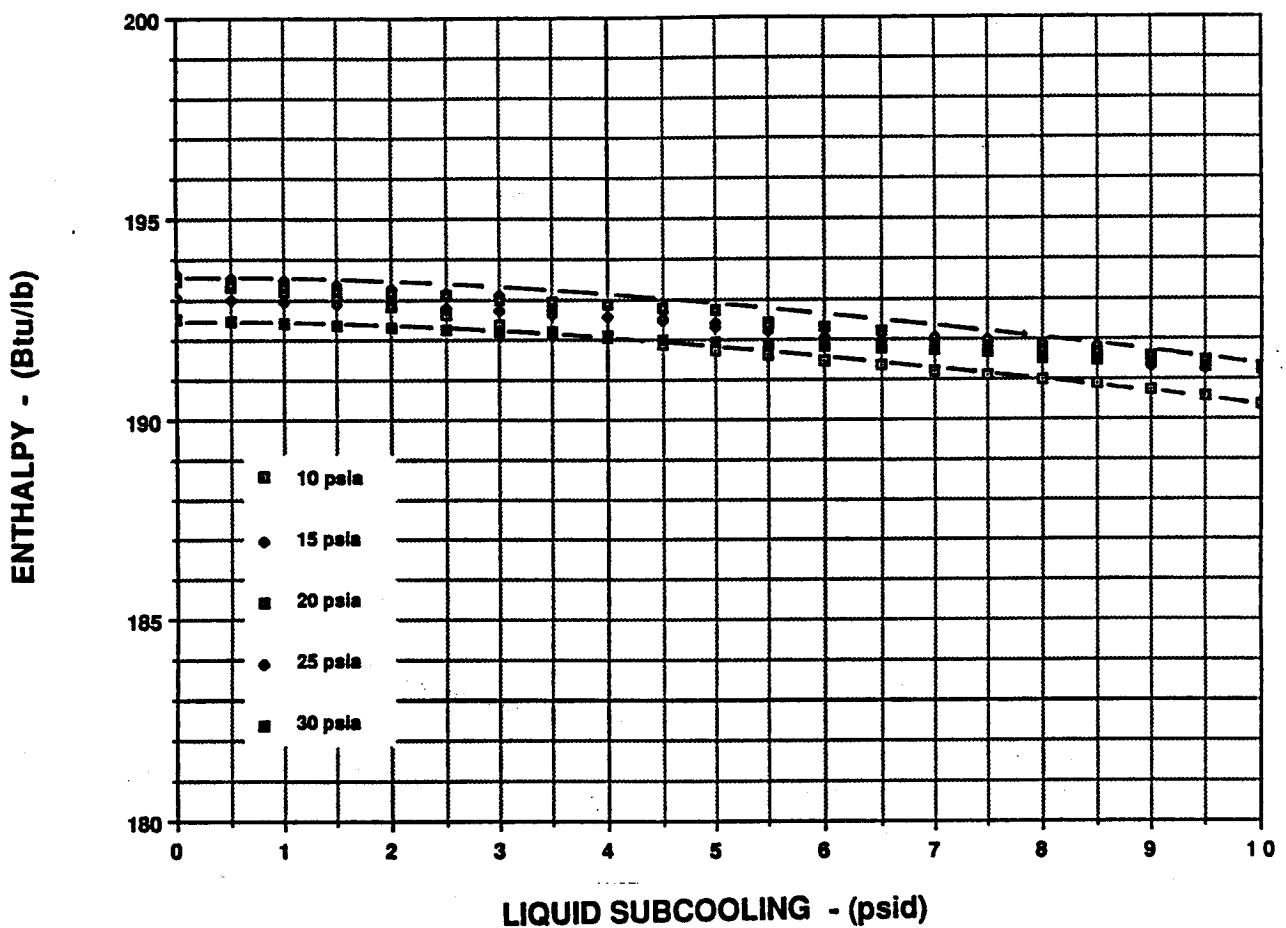


Figure 2.1.2 Heat Absorption Capability of Vent Flow as a Function of Liquid Subcooling

2.2.1 Spray Manifold

The spray manifold calculates the pressure drop through the manifold and determines the pressure at the inlet of the spray injection tubes (Fig. 2.2.1). The model accounts for frictional pressure drop in the manifold, and pressure losses resulting from flow turning and contraction at the exit of the manifold. From Bernoulli equation

$$\frac{(p_{SM})_i}{\rho} + \frac{V_{SM}^2}{2g_c} + az_i = \frac{(p_{SM})_o}{\rho} + \frac{V_{SM}^2}{2g_c} + az_o + (h_L)_{SM} \quad (2.2.1)$$

where $(p_{SM})_i$ is the spray manifold inlet pressure

$(p_{SM})_o$ is the spray manifold outlet pressure

V_{SM} is the velocity in the spray manifold

z_i, z_o are the inlet and outlet elevations

$a = \frac{g}{g_c}$ is the acceleration

The total head loss is defined as

$$(h_L)_{SM} = K_{SM} \frac{V_{SM}^2}{2g_c} \quad (2.2.2)$$

The total loss coefficient K_{SM} is given by

$$K_{SM} = (K_f)_{SM} + (K_b)_{SM} + (K_c)_{SM} \quad (2.2.3)$$

and includes

$$(K_f)_{SM} = f_{SM} \left(\frac{L}{D} \right)_{SM} \quad (\text{spray manifold frictional loss coefficient }) \quad (2.2.4a)$$

$$(K_b)_{SM} = f_{SM} \left(\frac{L_e}{D} \right) \quad (\text{90-degree bend resistance at the manifold exit }) \quad (2.2.4b)$$

$$(K_c)_{SM} = 0.5 \left[1 - \left(\frac{D_{SI}}{D_{SM}} \right)^2 \right] \quad (\text{sudden contraction at the manifold exit }) \quad (2.2.4c)$$

In Eqs. 2.2.4,

$L_{SM} = z_o - z_i$ is the spray manifold length

L_e is the bend equivalent length

D_{SI} is the spray injection tube ID

D_{SM} is the spray manifold ID

f_{SM} is the friction coefficient in the spray manifold obtained from

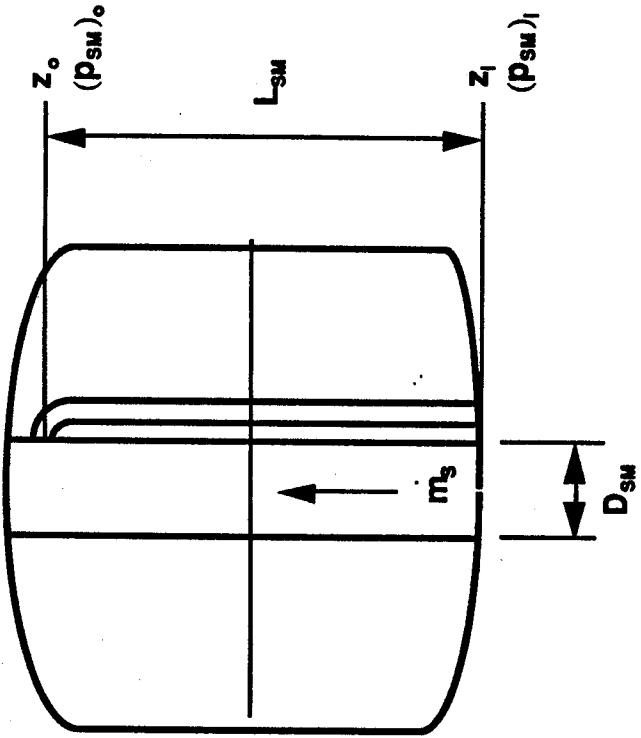


Figure 2.2.1 Spray Manifold Model

$$f_{SM} = \frac{1}{\left\{ 4 \log_{10} \left[\frac{\epsilon / D}{3.7} + \frac{2.51}{Re \sqrt{0.0056 + \frac{0.5}{(Re)^{0.32}}}} \right] \right\}^2}, Re > 3000$$

$$f_{SM} = \frac{64}{Re}, Re \leq 3000 \quad (2.2.5)$$

Eq. 2.2.1 can be solved for the spray manifold outlet pressure

$$(p_{SM})_o = (p_{SM})_i - K_{SM} q_{SM} - \rho a L_{SM} \quad (2.2.6)$$

where the dynamic pressure q_{SM} in the spray manifold is given by

$$q_{SM} = \rho \frac{V_{SM}^2}{2g_c} = \frac{1}{2\rho g_c} \left(\frac{\dot{m}_S}{A_{SM}} \right)^2 \quad (2.2.7)$$

2.2.2 Spray Injection Tube

The spray injection tube model is a multinode model which assigns a node to each orifice (Fig 2.2.2). Bernoulli equation is first applied to find the pressure downstream of the inlet 90 degree bend of the injection tube (pressure at the inlet of the straight section)

$$p_i = (p_{SM})_o - q_i (K_b)_{SI} \quad (2.2.8)$$

In Eq. 2.2.8, $(K_b)_{SI}$ is the 90 degree bend resistance and q_i is the inlet dynamic pressure given by

$$q_i = \frac{1}{2\rho g_c} \left(\frac{\dot{m}_i}{A_{SI}} \right)^2 \quad (2.2.9)$$

where A_{SI} is the flow area of an injection tube and \dot{m}_i is the mass flow rate in each tube (equal to the flow rate in the manifold divided by the number of tubes).

The straight section of the spray injection tube is divided into 45 equal nodes corresponding to the 45 spray orifices. Each node has a pressure and a mass flow rate at the inlet (i), center, and outlet (o) of the node. The outlet pressure and mass flow rate of one node is therefore the inlet pressure and mass flow rate of the preceding node

$$(p_i)_n = (p_o)_{n-1} \quad (2.2.9a)$$

$$\left(\dot{m}_i \right)_n = \left(\dot{m}_o \right)_{n-1} \quad (2.2.9b)$$

Bernoulli equation is applied successively from inlet to center, and from center to outlet to determine the pressure at the center and outlet of a node n.

From inlet to center,

$$p_n = (p_i)_n + \rho a \frac{\Delta z}{2} - K_f (q_i)_n \quad (2.2.10)$$

where Δz is the nodal length and K_f is the frictional loss coefficient.

From center to outlet,

$$(p_o)_n = p_n + \rho a \frac{\Delta z}{2} - K_f (q_o)_n \quad (2.2.11)$$

where the outlet dynamic pressure $(q_o)_n$ of node n is given by

$$(q_o)_n = \frac{1}{2\rho g_c} \left[\frac{\left(m_o \right)_n}{A_{SI}} \right]^2 \quad (2.2.12)$$

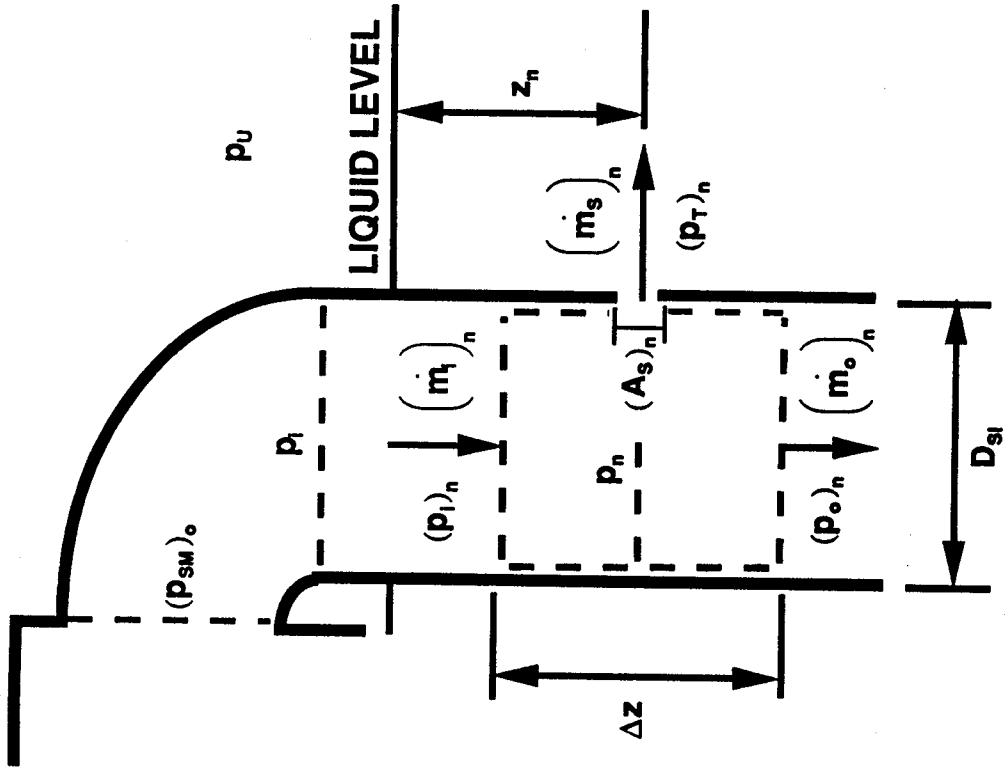


Figure 2.2.2 Spray Injection Tube Model

The mass flow rate at node n outlet $(\dot{m}_o)_n$ is obtained from

$$(\dot{m}_o)_n = (\dot{m}_i)_n - (\dot{m}_s)_n \quad (2.2.13)$$

$(\dot{m}_s)_n$ in Eq. 2.2.13 is the spray flow rate calculated from an incompressible flow relation

$$(\dot{m}_s)_n = (A_s)_n \sqrt{\frac{2\rho g_c [p_n - (p_T)_n]}{K_s}} \quad (2.2.14)$$

In Eq. 2.2.14, K_s is the loss coefficient of an orifice in a duct given by

$$K_s = \left[\frac{1}{C_d} - \frac{A_s}{A_T} \right]^2 \quad (2.2.15)$$

where C_d is the discharge coefficient ($C_d=0.8$), and A_s/A_T is the ratio of the orifice to the tank area ($A_s/A_T=0$). Thus, K_s is determined to be 1.56.

The tank pressure $(p_T)_n$ at node n is calculated as

$$\begin{aligned} (p_T)_n &= p_u && \text{(ullage nodes)} \\ &= p_u + \rho_L g z_n, && \text{(liquid nodes)} \end{aligned} \quad (2.2.16)$$

where z_n is the distance from the liquid surface to node n.

2.2.3 Spray Manifold and Injection Tube Model Algorithm

The flow chart of the spray manifold and injection tube model is given in Section 3.1.2. The model starts out with a guess of the pump flow rate and calculates the pressures and mass flow rates at each node. Knowing the pressure and spray flow rate of the last node N, it then calculates the tank pressure corresponding to that last node by solving the incompressible flow relation of Eq. 2.2.14

$$(p_T)_{N,calc} = p_N - \frac{K_s}{2\rho g} \sqrt{\frac{(\dot{m}_s)_N}{(A_s)_N}} \quad (2.2.17)$$

Next, $(p_T)_{N,calc}$ is compared with $(p_T)_N$ obtained from the ullage pressure and hydrostatic head (Eq. 2.2.16). If they are not equal within a specified tolerance (0.001 psi), a new guess of the pump flow rate will be made and the process repeated until convergence on $(p_T)_N$ is achieved.

2.3 Recirculation Pump Model

The zero-g TVS LH₂ recirculation pump is a centrifugal pump which is a constant output pressure device since it imparts kinetic pressure to the fluid due to rotation. Consequently, the pump pressure rise (Δp_p) is only a function of rotation speed (N) and tip velocity (U)

$$U = \frac{\pi D_m N}{720} \quad (2.3.1)$$

where D_m is the impeller diameter.

The fluid horsepower required by the pump flow (m), raised to Δp_p pressure, is equal to

$$HP_o = \frac{m \Delta p_p}{\eta_p \rho} \quad (2.3.2)$$

where η_p is the pump mechanical efficiency.

The pump operating speed then changes as a result of the energy absorbed by the fluid and the power supplied to the pump through a power source. The instantaneous rate of change in pump operating speed is

$$\frac{dN}{dt} = \left(\frac{HP_i - HP_o}{I_p N} \right) 6.0185 \times 10^5 \quad (2.3.3)$$

where I_p is the polar moment of inertia of the pump and HP_i is the input power to the pump.

Integration of the pump acceleration results in the pump speed at any given time

$$N = (N)_{ic} + \int \left(\frac{dN}{dt} \right) dt \quad (2.3.4)$$

By specifying the initial pump speed at zero, a pump start transient may be simulated.

A pump head-flow curve was provided by the pump manufacturer, Barber-Nichols Engineering Co. (Fig. 2.3.1). The curve was fitted with a polynomial function to give the head coefficient (ψ) as a function of the flow coefficient (ϕ)

$$\psi = 0.52889 - 1.4956\phi + 47.819\phi^2 - 485.93\phi^3 + 1633.9\phi^4 - 1833.5\phi^5 \quad (2.3.5)$$

The flow coefficient ϕ is obtained from test data in terms of the flow rate (in gpm) and the pump speed as

$$\phi = \frac{\text{gpm}}{0.0531N} \quad (2.3.6)$$

The pump head is calculated from the pump speed and head coefficient

$$H = 4.507 \times 10^{-6} N^2 \psi \quad (2.3.7)$$

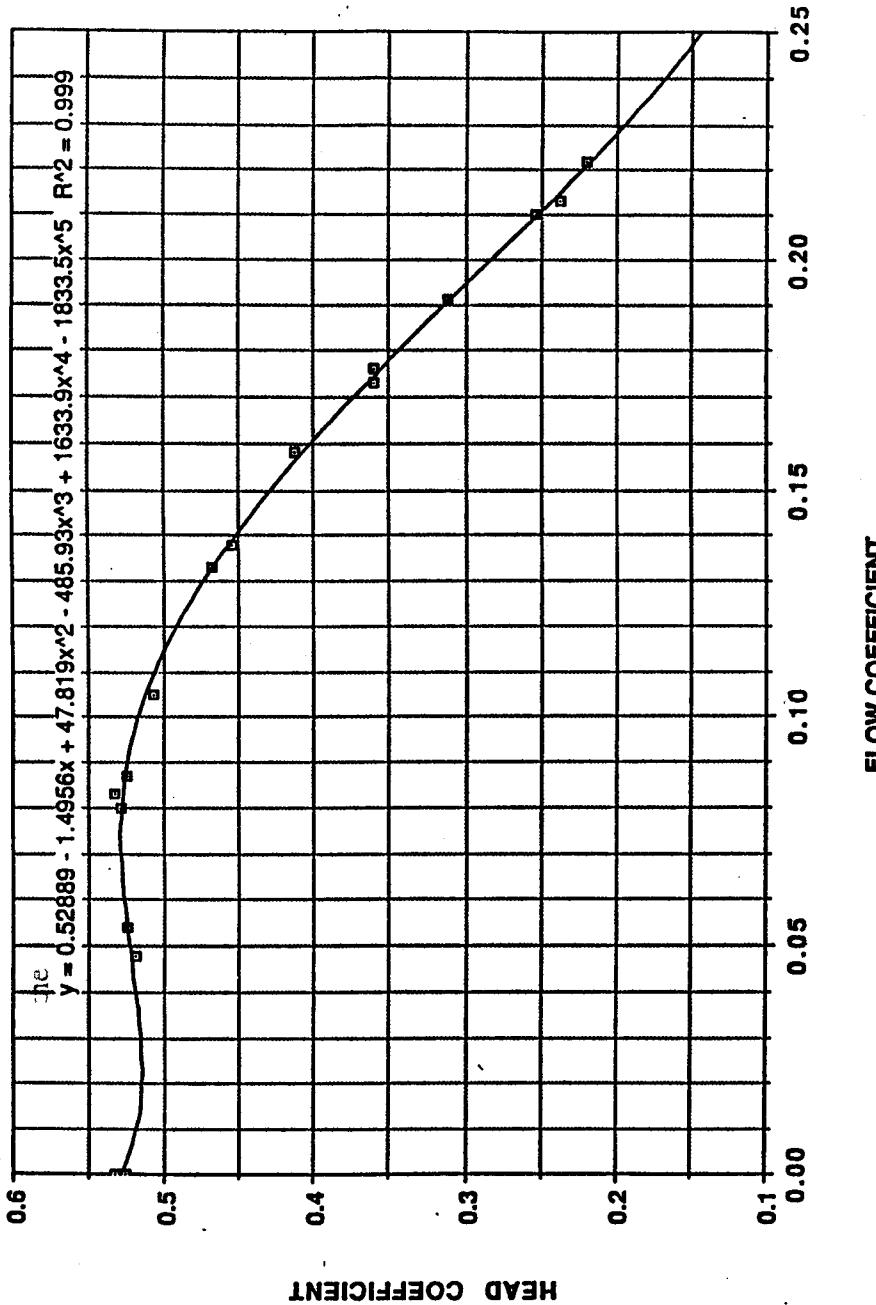


Figure 2.3.1 LH₂ Recirculation Pump Head-Flow Curve

The pump pressure rise is then obtained as

$$\Delta p_p = \frac{\rho H}{144} \quad (2.3.8)$$

The lumped pump model requires the pump design flow rate (Q_D) and speed (N_D) in order to define the other operating characteristics (HP_1 , I_p) required by the model.

2.4 Tank Thermal Model

The tank model is a lumped model consisting of four control volumes (Fig. 2.4.1): (1) ullage, (2) tank wall, (3) liquid on the tank wall, and (4) bulk liquid. The thermal model of each control volume is described in the following.

2.4.1 Ullage

The ullage thermal model applies conservation of mass and energy to determine the ullage pressure, temperature and mass (Fig. 2.4.2). From conservation of mass, the change in the ullage mass (M_U) is due to all masses entering and leaving the ullage control volume

- (1) droplet evaporation rate in the ullage (m_{DU})
- (2) boiling rate of the liquid on the tank wall (m_{bW})
- (3) bulk liquid boiling rate (m_{LU}), or ullage condensation (m_{UL})
- (4) liquid surface condensation (m_{COND})

$$\frac{dM_U}{dt} = m_{DU} + m_{bW} + m_{LU} - m_{UL} - m_{COND} \quad (2.4.1)$$

These mass flow rates are defined in Section 2.4.6. The ullage mass is obtained by integrating its time rate of change with respect to time

$$M_U = (M_U)_{IC} + \int \left(\frac{dM_U}{dt} \right) dt \quad (2.4.2)$$

From conservation of energy, the change in the ullage temperature (T_U) is the result of

- (1) heat transfer to the ullage (q_U)
- (2) work done on the ullage (w_U)
- (3) energy added to the ullage by incoming and leaving masses ($ENTH_U$)

$$\frac{dT_U}{dt} = \frac{q_U - w_U - ENTH_U - c_{vU} T_U \frac{dM_U}{dt}}{M_U c_{vU}} \quad (2.4.3)$$

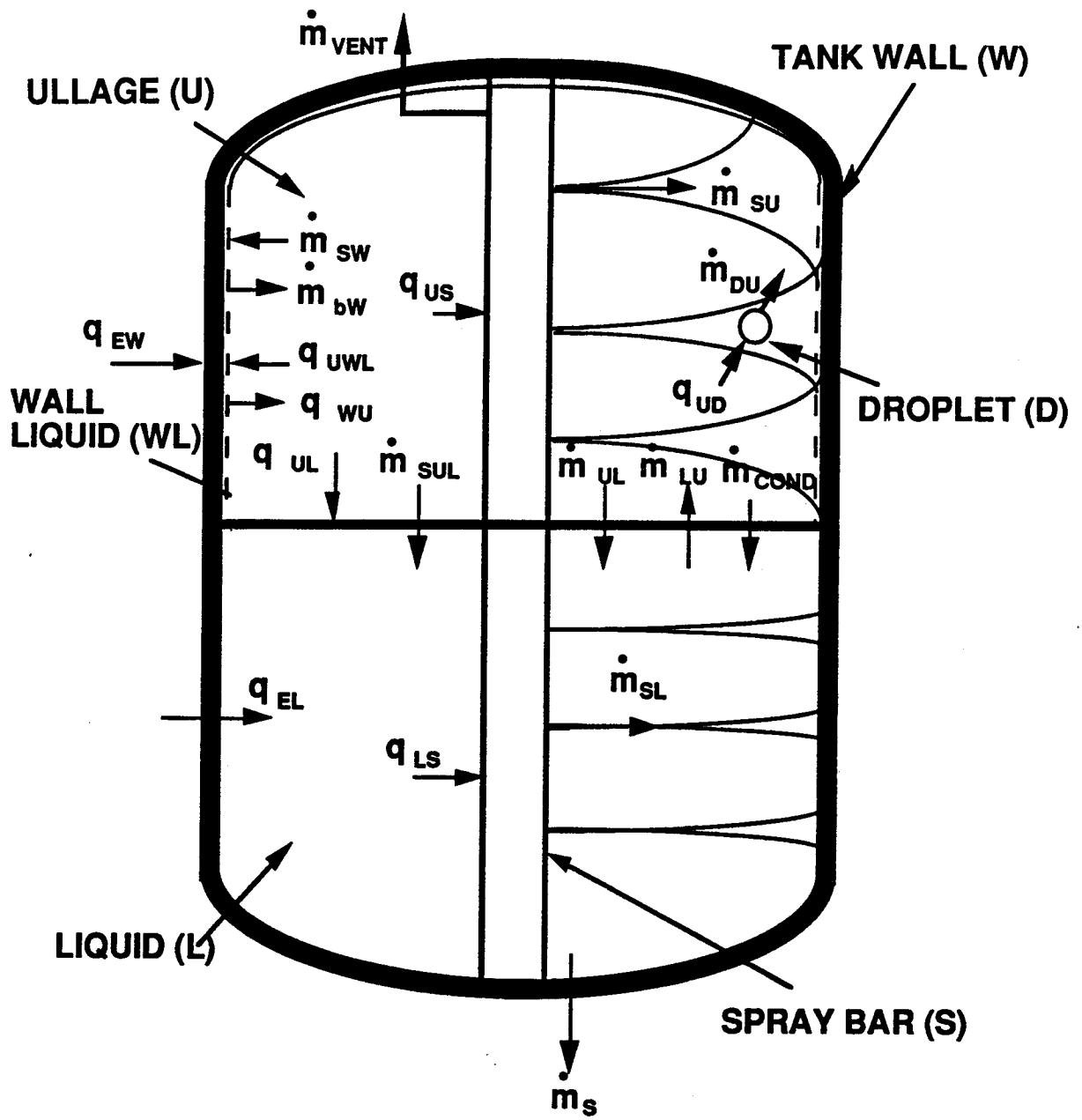


Figure 2.4.1 Tank Thermal Model

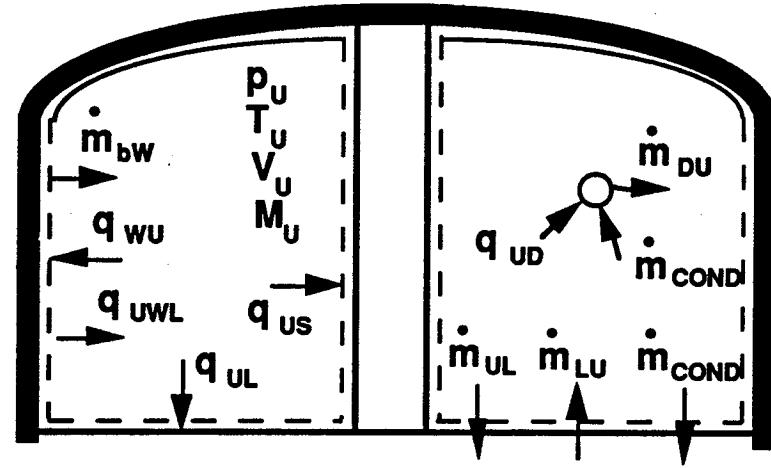


Figure 2.4.2 Ullage Thermal Model

The terms in Eq. 2.4.3 are defined as follows

$$(1) \quad q_u = q_{wu} - q_{uwL} - q_{uL} - q_{uD} - q_{us} \quad (\text{heat transfer to ullage})$$

where

q_{wu} is the heat-transfer rate between the tank wall and ullage,

$|q_{wu}| > 0$ for a dry wall

$= 0$ for a wet wall

q_{uwL} is the heat-transfer rate between the ullage and wall liquid,

$|q_{uwL}| = 0$ for a dry wall

> 0 for a wet wall

q_{uL} is the heat-transfer rate between the ullage and bulk liquid

q_{uD} is the heat-transfer rate between the ullage and liquid droplet

q_{us} is the heat-transfer rate between the ullage and (unsubmerged) spray bars

The above heat-transfer rates are defined in Section 2.4.5.

$$(2) \quad w_u = p_u \frac{dV_u}{dt} \quad (\text{work done on ullage})$$

where the change in the ullage volume ($\frac{dV_u}{dt}$) is equal and opposite to the change in the liquid and wall liquid volumes

$$\frac{dV_u}{dt} = -\frac{dV_L}{dt} - \frac{dV_{wL}}{dt} \quad (2.4.4)$$

$$(3) \quad ENTH_u = \left(\frac{dM_u}{dt} \right) h_{sat}$$

where $h_{sat} = h_{sat}(p_u)$ is the saturated vapor enthalpy of the ullage.

The ullage volume is obtained as the difference between the tank volume and the bulk liquid and wall liquid volumes

$$V_u = V_t - V_L - V_{wL} \quad (2.4.5)$$

Eq. 2.4.3 is integrated with respect to time to obtain the ullage temperature

$$T_u = (T_u)_{IC} + \int \left(\frac{dT_u}{dt} \right) dt \quad (2.4.6)$$

With the ullage mass, temperature and volume determined, the ullage pressure is calculated from the equation of state

$$p_u = \frac{M_u R_u T_u}{V_u} \quad (2.4.7)$$

2.4.2 Tank Wall

The tank wall is divided into two sections, one facing the liquid and the other facing the ullage. The tank wall facing the bulk liquid is assumed to be at the same temperature as the liquid. Thus, the tank wall thermal model described in this section applies to the section facing the ullage (Fig. 2.4.3). Since liquid can form on the tank wall as a result of spraying, the model must account for both dry and wet wall cases.

From conservation of energy, the change in the tank wall temperature is due to

- (1) heat input to the wall from the environment (q_{EW})
- (2) heat-transfer rate between the wall and ullage (q_{WU})

$$\begin{aligned} |q_{WU}| &> 0 \text{ for a dry wall} \\ &= 0 \text{ for a wet wall} \end{aligned}$$

- (3) heat-transfer rate between the wall and liquid on the wall (q_{WL}),

$$\begin{aligned} |q_{WL}| &= 0 \text{ for a dry wall} \\ &> 0 \text{ for a wet wall} \end{aligned}$$

$$\frac{dT_w}{dt} = \frac{q_{EW} - q_{WU} - q_{WL}}{M_w C_{pw}} \quad (2.4.8)$$

Section 2.4.5 defines these heat-transfer rates. Eq. 2.4.8 can be integrated with respect to time to obtain the tank wall temperature

$$T_w = (T_w)_{ic} + \int \left(\frac{dT_w}{dt} \right) dt \quad (2.4.9)$$

2.4.3 Wall Liquid

The wall liquid thermal model is also governed by the laws of conservation of mass and energy (Fig. 2.4.4). From conservation of mass, the change in the wall liquid mass (M_{WL}) is equal to the difference between the liquid mass reaching the wall and the liquid mass boiled off from the wall

$$\frac{dM_{WL}}{dt} = m_{sw} - m_{bw} \quad (2.4.10)$$

where m_{sw} is the spray flow rate reaching the wall and m_{bw} is the liquid boil-off rate from the wall.

These mass flow rates will be defined in Section 2.4.6. Eq. 2.4.10 can be integrated to obtain the wall liquid mass

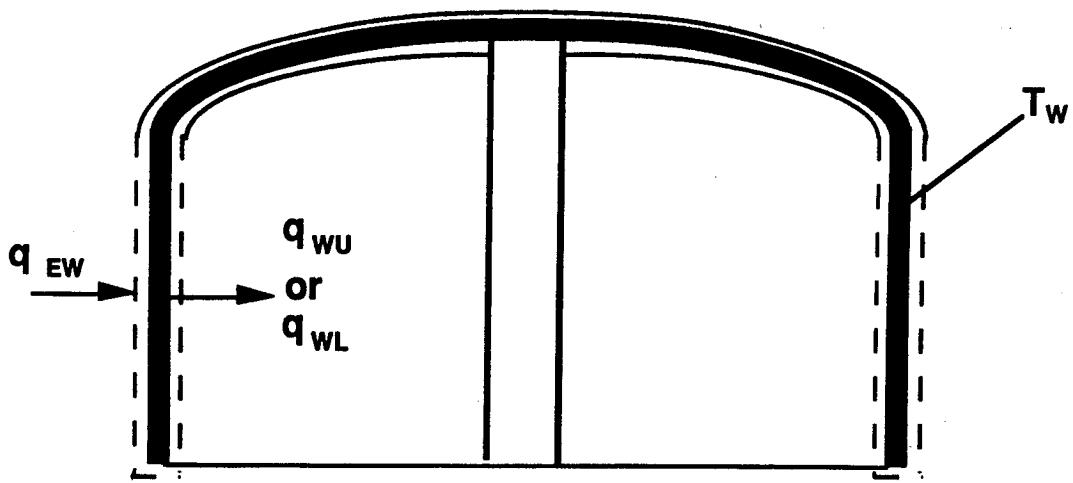


Figure 2.4.3 Tank Wall Thermal Model

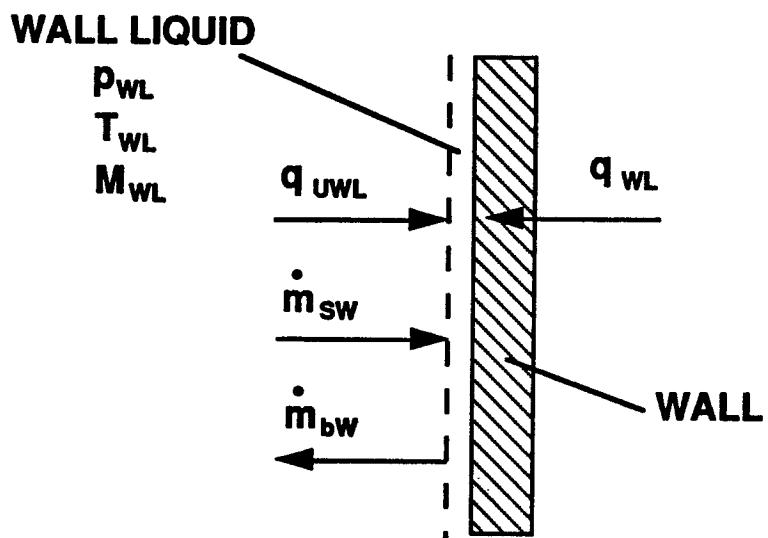


Figure 2.4.4 Wall Liquid Thermal Model

$$M_{WL} = (M_{WL})_{IC} + \int \left(\frac{dM_{WL}}{dt} \right) \quad (2.4.11)$$

From conservation of energy, the change in the wall liquid temperature (T_w) is the result of heat transfer to the wall liquid and sensible energy added to the spray to raise its temperature (T_{sw}) to the wall liquid temperature. Heat transfer to the wall liquid includes heat-transfer rate between the wall and wall liquid (q_{WL}), and heat-transfer rate between the ullage and wall liquid (q_{UWL}).

$$\frac{dT_{WL}}{dt} = \frac{q_{WL} + q_{UWL} - m_{sw} c_{pL} (T_{WL} - T_{sw})}{M_{WL} c_{pWL}} \quad (2.4.12)$$

These heat-transfer rates are defined in Section 2.4.5. Eq. 2.4.12 can be integrated to obtain the wall liquid temperature

$$T_{WL} = (T_{WL})_{IC} + \int \left(\frac{dT_{WL}}{dt} \right) dt \quad (2.4.13)$$

The wall liquid vapor pressure is then obtained from the thermodynamic data base as

$$p_{WL} = p_{sat}(T_{WL}) \quad (2.4.14)$$

The volume rate of change of the wall liquid is determined from Eq. 1.9 as

$$\frac{dV_{WL}}{dt} = \frac{1}{\rho_{WL}} \frac{dM_{WL}}{dt} \quad (2.4.15)$$

where $\rho_{WL} = \rho_{sat}(T_{WL})$ is the wall liquid density.

Eq. 2.4.15 is integrated to obtain the wall liquid volume

$$V_{WL} = (V_{WL})_{IC} + \int \left(\frac{dV_{WL}}{dt} \right) dt \quad (2.4.16)$$

2.4.4 Bulk Liquid

Originally conceived as multi-node, the bulk liquid thermal model is made single node since (1) mixing will destratify the liquid and create a uniform bulk, and (2) uncertainty in heat-transfer modeling does not justify the added complexities of a multinode model.

The liquid thermal model is also based on the laws of conservation of mass and energy. From conservation of mass, the change in the liquid mass must be balanced by a change in the ullage mass and any mass vented overboard (Fig. 2.4.5).

$$\frac{dM_L}{dt} = m_{SL} + m_{SUL} + m_{COND} + m_{UL} - m_{LU} - m_s - m_v \quad (2.4.17)$$

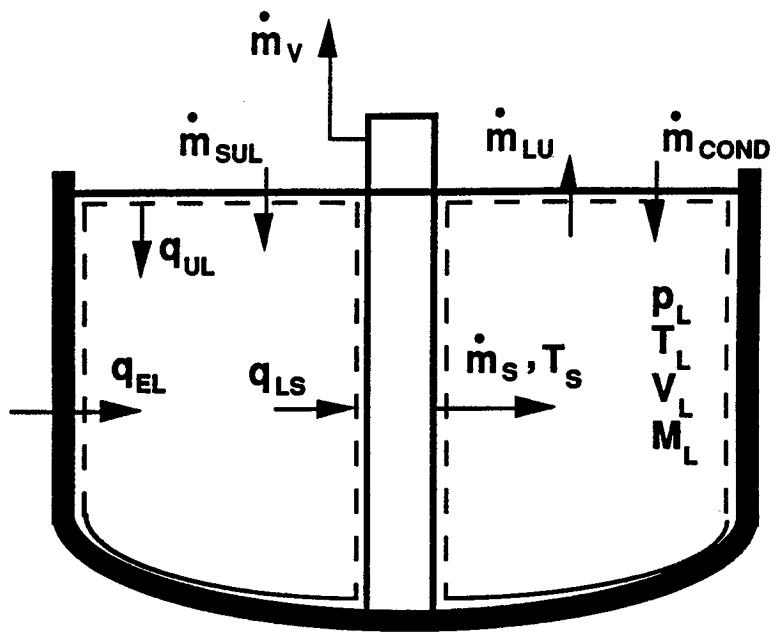


Figure 2.4.5 Bulk Liquid Thermal Model

where m_{SL} is the liquid spray flow rate into the bulk liquid

m_{SUL} is the unevaporated droplet flow rate

m_{COND} is the liquid surface condensation flow rate

m_{UL} is the ullage condensation flow rate

m_{LU} is the liquid boil-off rate

m_s the pump flow rate

m_v is the overboard venting flow rate.

The liquid mass is obtained by integrating its time rate of change

$$M_L = (M_L)_{IC} + \int \left(\frac{dM_L}{dt} \right) \quad (2.4.18)$$

From conservation of energy, the change in the liquid temperature is caused by

- (1) heat transfer to the liquid
- (2) heat added by the unevaporated droplets
- (3) sensible energy added to the liquid spray to raise its temperature (T_s) to the liquid temperature
- (4) latent heat of vaporization of the liquid

$$\frac{dT_L}{dt} = \frac{q_L + m_{SUL} c_{pL} (T_d - T_L) - m_{LU} (h_{fg})_L - m_{SU} c_{pL} (T_L - T_s)}{M_L c_{pL}} \quad (2.4.19)$$

The heat-transfer rate to the liquid (q_L) is given by

$$q_L = q_{EL} + q_{UL} - q_{LS}$$

where q_{EL} is the heat added to the liquid by the environment

q_{UL} is the heat-transfer rate between the ullage and liquid

q_{LS} is the heat-transfer rate between the liquid and (submerged) spray bars

These heat-transfer rates are given in Section 2.4.5. Eq. 2.4.19 is integrated with respect to time to give the liquid temperature

$$T_L = (T_L)_{IC} + \int \left(\frac{dT_L}{dt} \right) dt \quad (2.4.20)$$

The liquid vapor pressure is obtained from the thermodynamic data base as

$$p_L = p_{sat}(T_L) \quad (2.4.21)$$

The liquid volume rate of change is determined from the rate of change of the liquid mass

$$\frac{dV_L}{dt} = \frac{1}{\rho_L} \frac{dM_L}{dt} \quad (2.4.22)$$

where $\rho_L = \rho_{sat}(T_L)$ is the liquid density.

Eq. 2.4.22 is integrated to give the liquid volume

$$V_L = (V_L)_{ic} + \int \left(\frac{dV_L}{dt} \right) dt \quad (2.4.23)$$

2.4.5 Heat Transfer

This section defines the heat-transfer rates which are found in the energy balances of Section 2.4.1 to 2.4.4. These heat-transfer rates can be divided into two groups: free convection and forced convection. Free convection is the dominant heat-transfer mode in the ullage and liquid, while forced convection characterizes liquid droplet heat transfer in the ullage.

The convection heat-transfer rate is generally defined as

$$q = hA\Delta T$$

where h is the convection heat-transfer coefficient

A is the surface area of heat transfer

ΔT is the temperature difference between the heat source and sink

The heat-transfer coefficient is obtained from the Nusselt Number (Nu) as

$$h = \left(\frac{k_F}{L_c} \right) Nu$$

where k_F is the fluid thermal conductivity and L_c is the surface characteristic length.

The Nusselt number is a function of the Rayleigh number (Ra) defined as

$$Ra = \frac{a\beta\Delta TL_c^3\rho^2c_p}{\mu k} \quad (2.4.24)$$

where a is the acceleration

β is the thermal expansion coefficient,

$$\beta = \frac{1}{T_f} \text{ for gas, } \frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p \text{ for liquid}$$

L_c is the characteristic length

ρ is the density

c_p is the specific heat at constant pressure

μ is the dynamic viscosity

k is the thermal conductivity

All properties must be evaluated at the film temperature (T_f) which is defined as the average of the fluid and surface temperatures.

2.4.5.1 Free Convection

Two free convection heat-transfer correlations are used in the model. The first one is a free convection correlation for interior surfaces of vertical ducts, vertical plates and cylinders, and horizontal cylinders (Ref. 28) (Fig 2.4.6)

$$Nu = 0.555Ra^{0.25} + 0.447 \quad (2.4.25)$$

This correlation is used to calculate the heat-transfer coefficients

- (1) between the ullage and wall (h_{uw})
- (2) between the ullage and bulk liquid (h_{ul})
- (3) between the ullage and wall liquid (h_{uwl})
- (4) between the ullage and (unsubmerged) spray bars (h_{us})
- (5) between the bulk liquid and (submerged) spray bars (h_{ls})

The characteristic length for h_{uw} , h_{ul} and h_{uwl} is the internal tank diameter while that of h_{us} and h_{ls} is the spray bar diameter.

The second correlation is the McAdams correlation for free convection of vertical surfaces in the turbulent range (Ref. 17)

$$Nu = 0.13Ra^{1/3} \quad (2.4.26)$$

This correlation is used to calculate the heat-transfer coefficient between the wall and wall liquid (h_{wl}). Because of the 1/3 power in Ra, h_{wl} can be obtained without knowing the characteristic length, thereby removing the uncertainty in determining the wall liquid layer.

2.4.5.2 Forced Convection

The forced convection heat-transfer coefficient between the ullage and liquid droplets (h_{ud}) is based on a McAdams recommended correlation for flow over a sphere (Ref. 8) (Fig. 2.4.7)

$$Nu = 0.3125Re^{0.602} \quad (2.4.27)$$

The Reynolds number of the spray flow (Re) is defined as

$$Re = \frac{\rho Vel_D D_D}{\mu} \quad (2.4.28)$$

where Vel_D is the droplet velocity in the ullage

D_D is the droplet diameter assumed to be equal to the orifice diameter
 ρ, μ are the density and viscosity of the ullage gas

Since the droplet diameter and velocity vary with the orifice size, the droplet heat-transfer

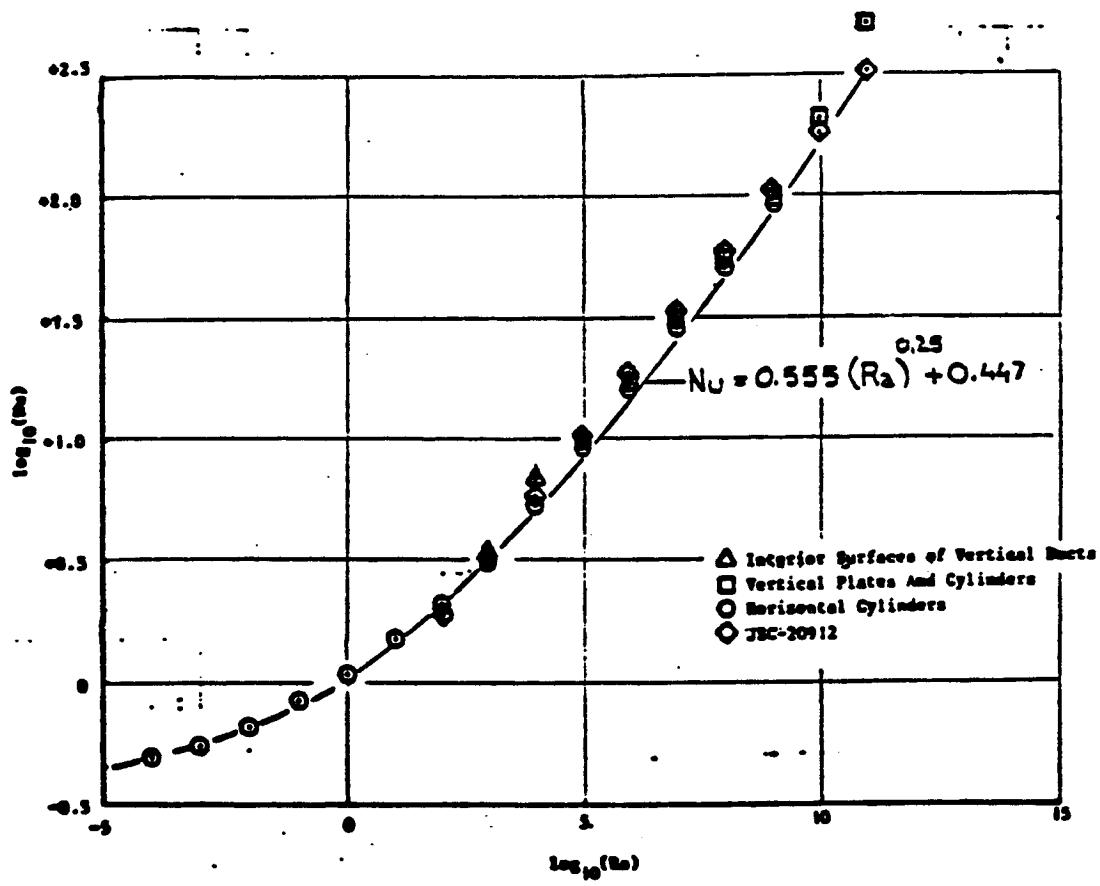


Figure 2.4.6 Free Convection Heat-Transfer Correlation for Interior Surfaces of Vertical Ducts, Vertical Plates and Cylinders, and Horizontal Cylinders

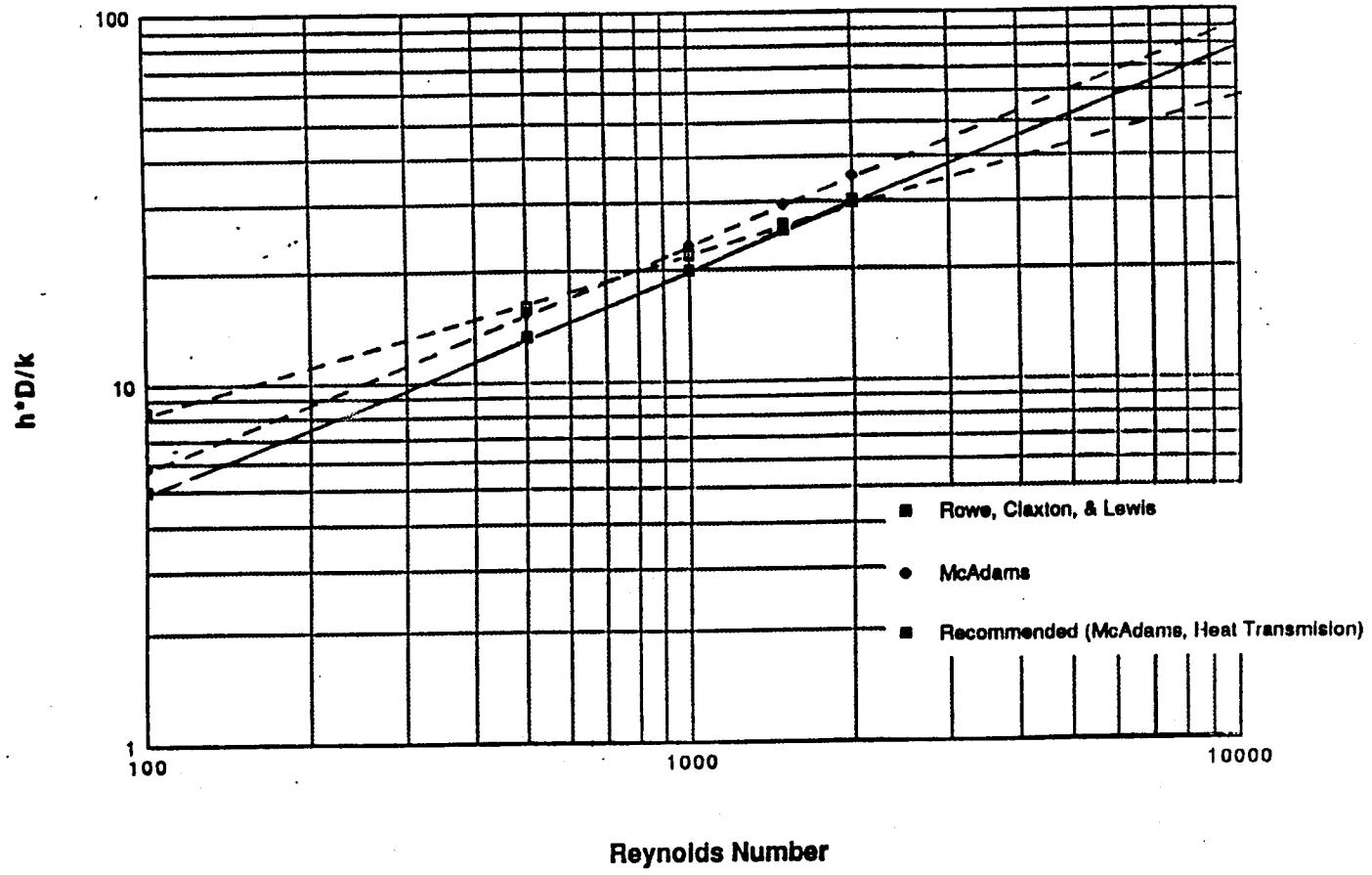


Figure 2.4.7 Forced Convection Heat-Transfer Correlation for Flow Over a Sphere

coefficient must be determined for each orifice. The total droplet heat-transfer rate is obtained by summing the droplet heat-transfer rates from each orifice

$$q_{\text{drop}} = \sum_{i=1}^n (n_{\text{drop}})_i (q_{\text{drop}})_i \quad (2.4.29)$$

where $(n_{\text{drop}})_i$ is the number of droplets sprayed from orifice i into the ullage. This is given by

$$(n_{\text{drop}})_i = \frac{\left(\dot{m}_{\text{su}}\right)_i D_{\text{CHAR}}}{2\rho_D (V_D)_i (Vel_D)_i} \quad (2.4.30)$$

where $\left(\dot{m}_{\text{su}}\right)_i$ is the spray flow rate into the ullage from orifice i

$(V_D)_i$ and $(Vel_D)_i$ are the droplet volume and velocity from orifice i

ρ_D is the droplet density

D_{CHAR} is a characteristic length determined empirically.

By correlating the zero-g TVS model with LeRC ullage pressure collapse data, this characteristic length was determined to be 1/4 of the tank diameter.

2.4.6 Mass Transfer

This section defines the mass-transfer rates found in the mass balance equations of Section 2.4.1 to 2.4.4 which include

- (1) Bulk liquid boiling (\dot{m}_{LU})
- (2) Liquid boiling from the tank wall (\dot{m}_{bw})
- (3) Liquid droplet evaporation in the ullage (\dot{m}_{DU})
- (4) Liquid spray falling into the bulk liquid (\dot{m}_{SL}) or accumulating on the tank wall (\dot{m}_{sw})
- (5) Ullage condensation (\dot{m}_{UL})
- (6) Liquid surface condensation (\dot{m}_{COND})

2.4.6.1 Bulk Liquid Boiling

Bulk liquid boiling occurs when the liquid vapor pressure is equal to the tank ullage pressure. It can be the result of heat transfer to the liquid and/or pressure decay in the ullage. It must also include sensible energy added to the liquid spray to increase its temperature to the liquid temperature.

If $p_L = p_U$,

$$\dot{m}_{\text{LU}} = \frac{1}{(h_{fg})_L} \left[q_L - \dot{m}_{\text{SL}} c_p (T_L - T_s) \right], \quad \frac{dp_U}{dt} < 0$$

$$= \frac{1}{(h_{fg})_L} \left[q_L - m_{SL} c_{pL} (T_L - T_s) - M_L c_{pL} \left(\frac{\partial T}{\partial p} \right)_{sat} \left(\frac{dp_U}{dt} \right) \right], \quad \frac{dp_U}{dt} > 0 \quad (2.4.31)$$

A polynomial fit of the LH₂ saturation temperature vs. pressure curve was obtained and its derivative taken to give an expression for $\left(\frac{\partial T}{\partial p} \right)_{sat}$

$$\left(\frac{\partial T}{\partial p} \right)_{sat} = 0.37781 - 4.9170 \times 10^{-3} p_L + 21.7623 \times 10^{-6} p_L^2 \quad (2.4.32)$$

If the ullage pressure increases above the liquid vapor pressure, boiling stops
 $m_{LU} = 0$, if $p_L < p_U$.

2.4.6.2 Wall Liquid Boiling

Wall liquid boiling from the tank wall follows the same mechanism as bulk liquid boiling

If $p_{WL} = p_U$,

$$m_{bw} = \frac{1}{(h_{fg})_L} \left[q_{WL} + q_{UWL} - m_{sw} c_{pL} (T_{WL} - T_{sw}) \right], \quad \frac{dp_U}{dt} < 0$$

$$= \frac{1}{(h_{fg})_L} \left[q_{WL} + q_{UWL} - m_{sw} c_{pL} (T_{WL} - T_{sw}) - M_{WL} c_{pL} \left(\frac{\partial T}{\partial p} \right)_{sat} \left(\frac{dp_U}{dt} \right) \right], \quad \frac{dp_U}{dt} > 0 \quad (2.4.33)$$

$$\text{where } \left(\frac{\partial T}{\partial p} \right)_{sat} = 0.37781 - 4.9170 \times 10^{-3} p_{WL} + 21.7623 \times 10^{-6} p_{WL}^2 \quad (2.4.34)$$

If $p_{WL} < p_U$, $m_{bw} = 0$.

As with bulk boiling, wall liquid boiling includes heat transfer to the wall liquid and sensible energy added to the spray liquid to bring its temperature to the wall liquid temperature.

2.4.6.3 Liquid Droplet Evaporation in the Ullage

Liquid droplets in the ullage will start boiling once the subcooled liquid spray is brought to saturation. From an energy balance on the liquid droplets, an expression for the liquid droplet boiling is obtained

$$m_{DU} = \frac{1}{(h_{fg})_U} \left[q_{UD} - m_{su} c_{pL} (T_{Usat} - T_s) \right] \quad (2.4.35)$$

where $T_{Usat} = T_{sat}(p_U)$ is the ullage saturation temperature.

2.4.6.4 Liquid Spray Falling into the Bulk Liquid or Accumulating on the Tank Wall

The unevaporated sprayed mass in the ullage is assumed to fall into the bulk liquid under 1 g, or to accumulate on the tank wall under 0 g (Fig. 2.4.8), i.e.,

$$m_{SUL} = m_{SU} - m_{DU} \quad (\text{for } 1 \text{ g}) \quad (2.4.36)$$

$$m_{SW} = m_{SU} - m_{DU} \quad (\text{for } 0 \text{ g})$$

2.4.6.5 Ullage Condensation

Ullage condensation occurs whenever the ullage temperature is equal to the saturation temperature corresponding to the ullage pressure. It is the result of heat removal from the liquid droplet (when there is spraying) and the wall liquid (Figure 2.4.9)

$$m_{UL} = \frac{q_{UD} + q_{UL} + q_{UWL}}{(h_{fg})_U} \quad T_U = T_{sat}(p_U) \quad (2.4.37)$$

2.4.6.6 Liquid Surface Condensation

When helium is not present to act as a barrier to mass transfer, bulk liquid mixing during pump operation induces condensation on the liquid surface. This condensation rate is controlled by the heat transfer rate from the ullage to the liquid

$$m_{COND} = \frac{q_{UL}}{(h_{fg})_U} \quad (2.4.38)$$

2.5 References

1. Chato, D. J., "Analysis of the Nonvented Fill of a 4.96-Cubic-Meter Lightweight Liquid Hydrogen Tank," NASA TM-102039.
2. Chen, I. M., "Thermodynamic Model for On-Orbit Cryogen Transfer," Summary Report, Rockwell International, IR&D Project 28601, September 1982.
3. Chen, I. M. and Moses, R. A., "A Shuttle Flight Experiment for On-Orbit Transfer of Cryogenic Hydrogen," AIAA 20th Joint Propulsion Conference, June 1984.
4. Chen, J. C., "A Correlation for Boiling Heat Transfer to Saturated Fluids in Convective Flow," ASME Paper No. 63-HT-34, ASME-AIChE Heat Transfer Conference and Exhibit, Boston, MA, August 1963.
5. Fischer, M. F., Schmidt, G. R., and Martin, J. J., "Analysis of Cryogenic Propellant Behavior in Microgravity and Low Thrust Environments," AIAA Paper 91-2173, June 1991.
6. Hubbard, G. L., Denny, V. E., and Mills, A. F., "Droplet Evaporation: Effects of Transients and Variable Properties," Int. J. Heat Mass Transfer, Vol. 18, pp. 1003-1008, 1975.

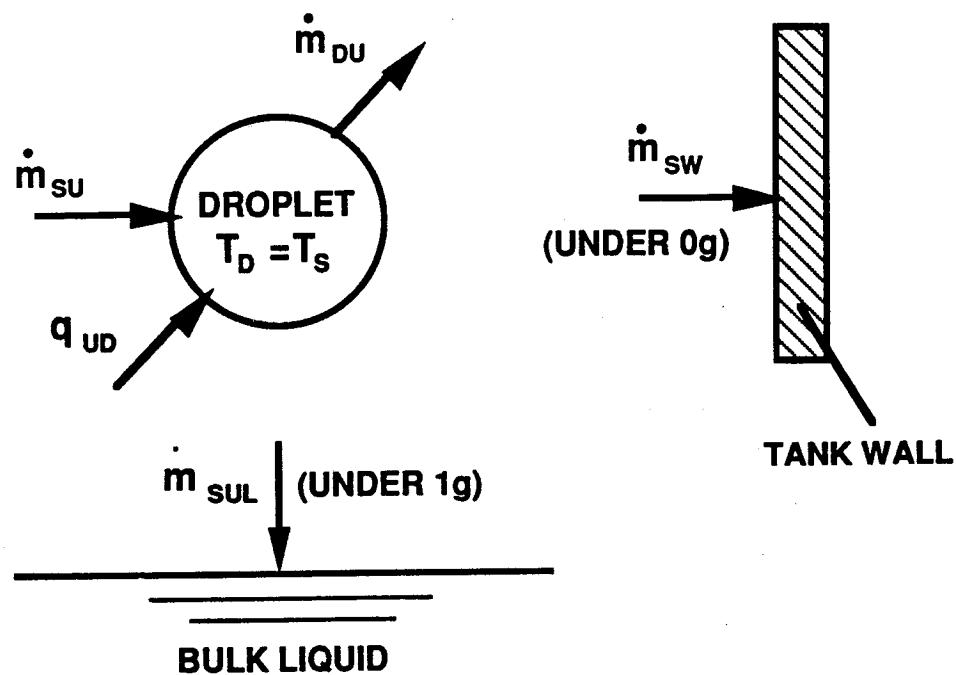


Figure 2.4.8 Droplet Evaporation Model

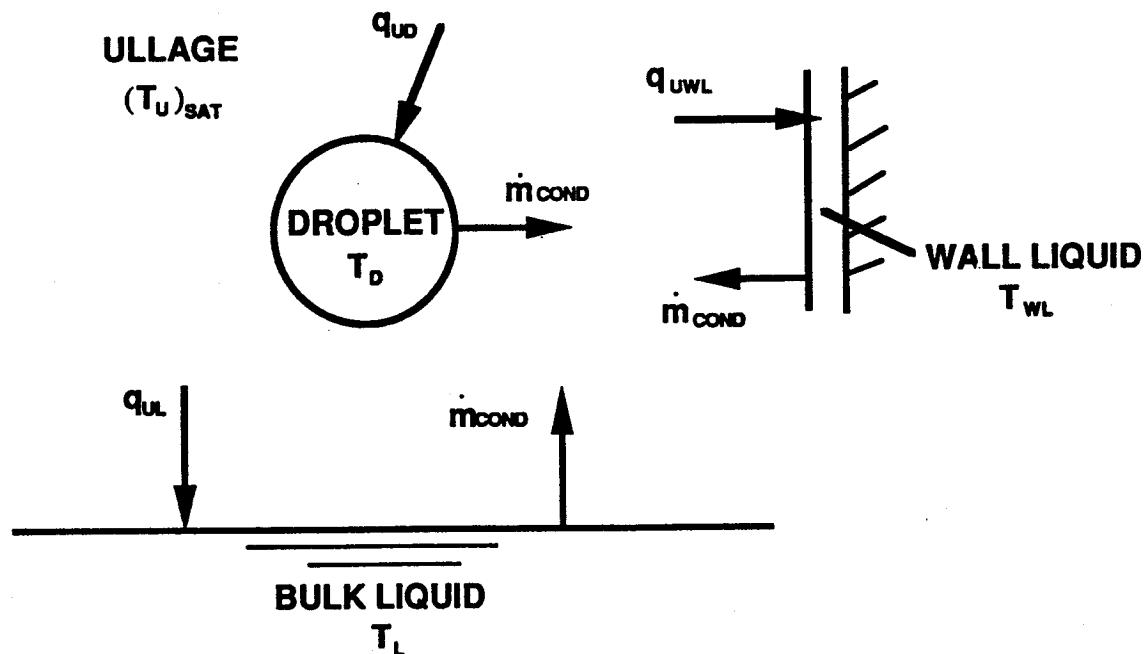


Figure 2.4.9 Ullage Condensation Model

7. Ingebo, R. D., "Vaporization Rates and Heat Transfer Coefficients for Pure Liquid Droplets," NASA TN 2368, July 1951.
8. Kreith, H., "Principles of Heat Transfer," 3rd edition, New York, 1973.
9. Lebedev, P. I., Rybanin, S. S., and Stesik, L. N., "Heat and Mass Transfer of a Liquid Drop Evaporating into a Flow of Gas Capable of Condensing," Int. J. Heat Mass Transfer, Vol. 24, No. 4, pp. 611-620, 1981.
10. Lefebvre, A. H., "Gas Turbine Combustion," Hemisphere Publishing Corporation, 1983.
11. Lin, C.-S., Van Dresar, N. T. and Hasan, M. M., "A Pressure Control Analysis of Cryogenic Storage Systems," AIAA Paper 91-2405, June 1991.
12. Miller, W. S., Kalvinskas, L. A., and Seader, J. D., "Boiling Heat Transfer for Cryogenics," NASA Report CR-243, 1965.
13. Moran, M. E. and Nyland, T. W., "Hydrogen No-Vent Fill Testing in a 5 Cubic Foot (142 Liter) Tank Using Spray Nozzle and Spray Bar Liquid Injection," AIAA Paper 92-3063, June 1992.
14. Ostroumov, G. A., "Free Convection Under the Conditions of the Internal Problem," State Publishing House, Technico-Theoretical Literature, Moscow-Leningrad, 1952, NACA TM 1407.
15. Ranz, W. E. and Marshall, W. R., "Evaporation from Drops, Parts I," Chemical Engineering Progress, Vol. 48, No. 3. pp. 141-146, March 1952.
16. Ranz, W. E. and Marshall, W. R., "Evaporation from Drops, Parts II," Chemical Engineering Progress, Vol. 48, No. 4. pp. 173-180, April 1952.
17. Ring, E., "Rocket Propellant and Pressurization Systems," Prentice-Hall, Inc., Englewood Cliff, N. J.
18. Sabnis, J. S. and de Jong, F. J., "Calculation of the Two-Phase Flow in an Evaporating Spray Using an Eulerian-Lagrangian Analysis," AIAA Paper 90-0447, January 1990.
19. Schmidt, G. R., Carrigan, R. W., Hahs, J. E., Vaughan, D. A., and Foust, D. C., "No-Vent Fill Pressurization Tests Using a Cryogen Simulant," NASA TM-103561.
20. Schmidt, G. and Hastings, L., "Cryogenic Fluid Management at MSFC," AIAA Paper 90-3711, September, 1990.
21. Taylor, W. J. and Chato, D. J., "Comparing the Results of an Analytical Model of the No-Vent Fill Process With No-Vent Fill Test Results for a 4.96 m^3 (175 ft 3) Tank," AIAA Paper 92-3078, July 1992.
22. Taylor, W. J. and Chato, D. J., "Improved Thermodynamic Modeling of the No-Vent Fill Process and Correlation With Experimental Data," NASA TM-104492.
23. Taylor, W. J., Chato, D. J., Moran, M. M., and Nyland, T. W., "On-Orbit Cryogenic Fluid Transfer Research at NASA Lewis Research Center," Cryogenics, Vol. 32, No. 2, 1992.

24. Van Drasar, N. T., "Pressurization of Cryogens: A Review of Current Technology and Its Applicability to Low-Gravity Conditions," AIAA Paper 92-3061, July 1992.
25. Vaughan, D. A. and Schmidt, G. R., "Analytical Modeling of No-Vent Fill Process," J. Spacecraft, Vol. 28, No. 5, pp. 574-579, Sept.-Oct. 1991.
26. Yuen, M. C. and Chen, L. W., "Heat Transfer Measurements of Evaporating Liquid Droplets," Int. J. of Heat Mass Transfer," Vol. 21, pp. 537-542, 1978.
27. "COOLANT: The Cryogenic On-Orbit Liquid Analytical Tool User's Manual," Vol. I, Version 2.0, General Dynamics Report No. GDSS-CRAD-88-005, Rev. A, October 1989.
28. "Generic Fluid Transfer Model, Computer Model Description Document," Rockwell International, December 1987.
29. "MPS Propellant Dump and Vacuum Inerting (The Generalized Two-Phase Cryogenic Propellant Dump Model)," Program No. MPS-17, 287-104-91-MPS-017, Rockwell International, June 1991.
30. "SAE Aerospace Applied Thermodynamics Manual," Developed by SAE Committee AC-9, Aircraft Environmental System.

2.6 Sample Cases

The sample cases shown are for a tank with 90% and 25% liquid quantities and a 0.25 Btu/hr-ft² heat flux. The tank is the 639-ft³ Multi-purpose Hydrogen Test Bed (MTHB) tank which is a cylindrical tank with elliptical bulkheads at both ends. The cylinder measures 5 ft in length and 10 ft in diameter while the bulkhead has a height of 2.5 ft. The tank has a wall thickness of 0.5 in and is made of aluminum. One-g acceleration is assumed and no helium is present in the tank. The results show the ullage and liquid vapor pressures, recirculation and vent flow rates, time between destratification and venting, destratification time, and TVS operation frequency.

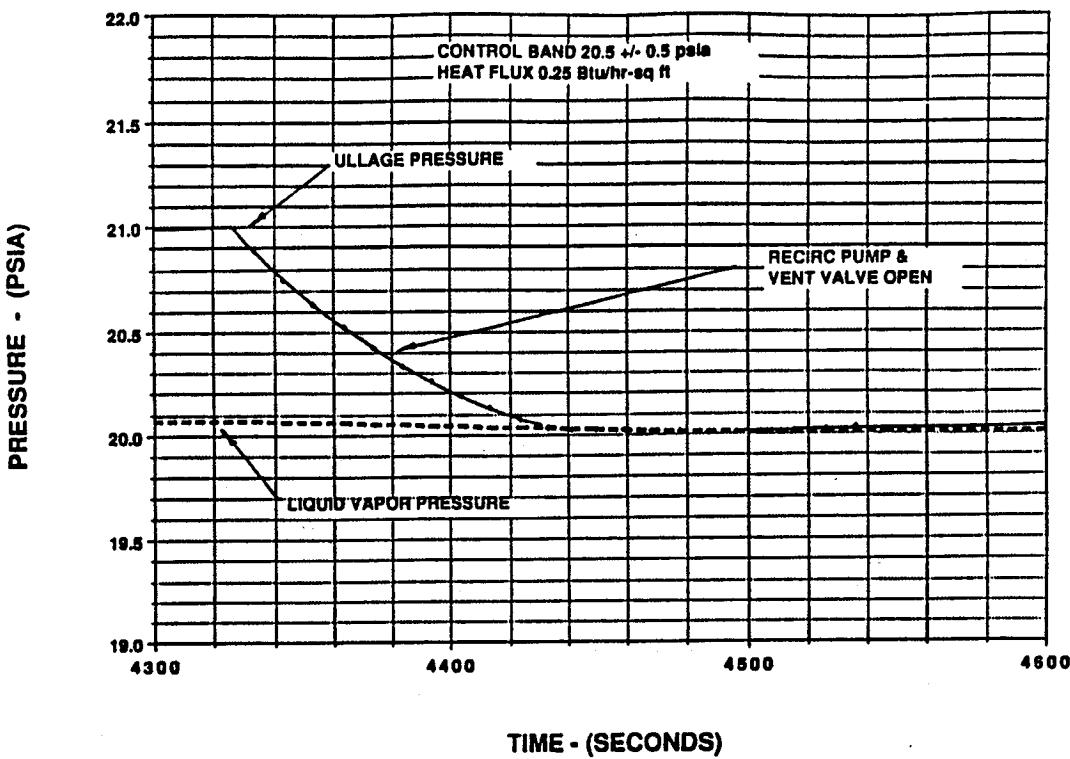


Figure 2.6.1 TVS Performance Simulation at 90% Liquid Quantity

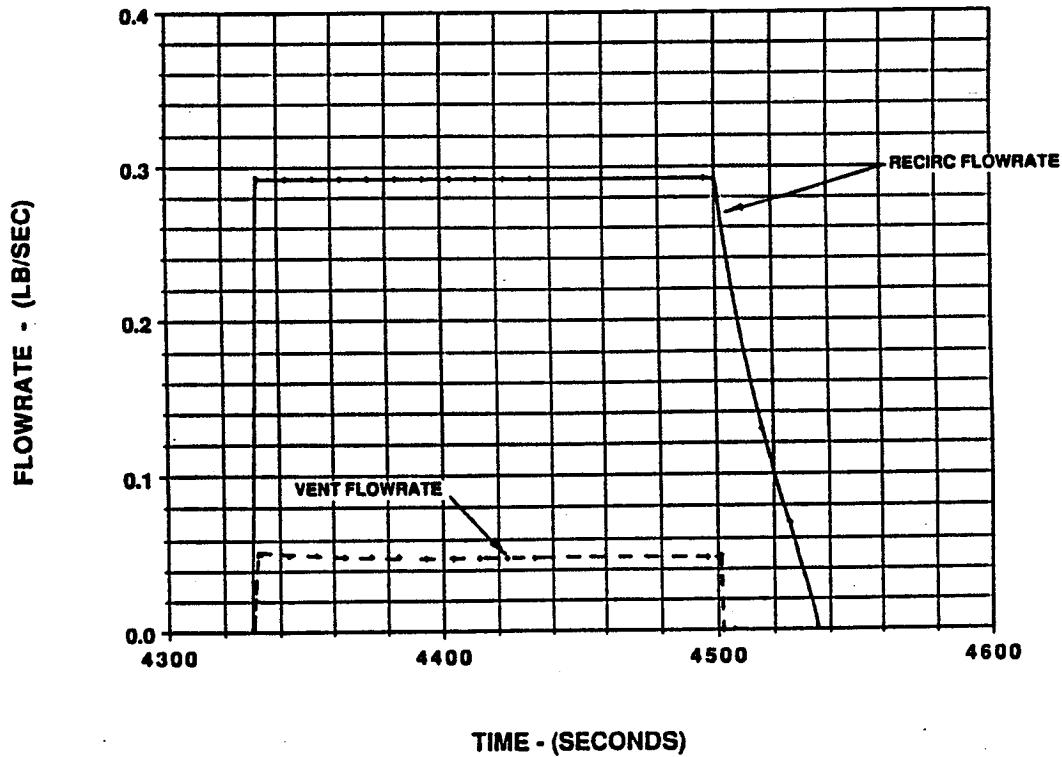


Figure 2.6.2 TVS Recirculation Pump and Vent Valve Flow Rate Transient During Ullage Destratification (90% Liquid Quantity)

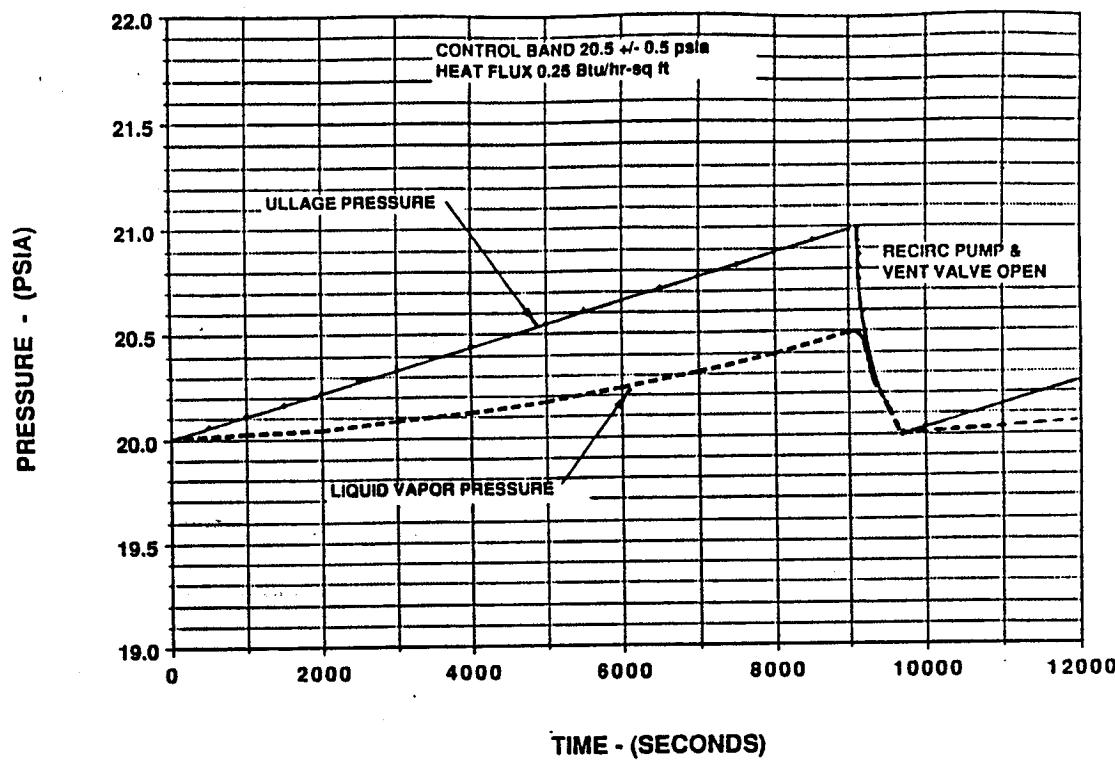


Figure 2.6.3 TVS Performance Simulation at 25% Liquid Quantity

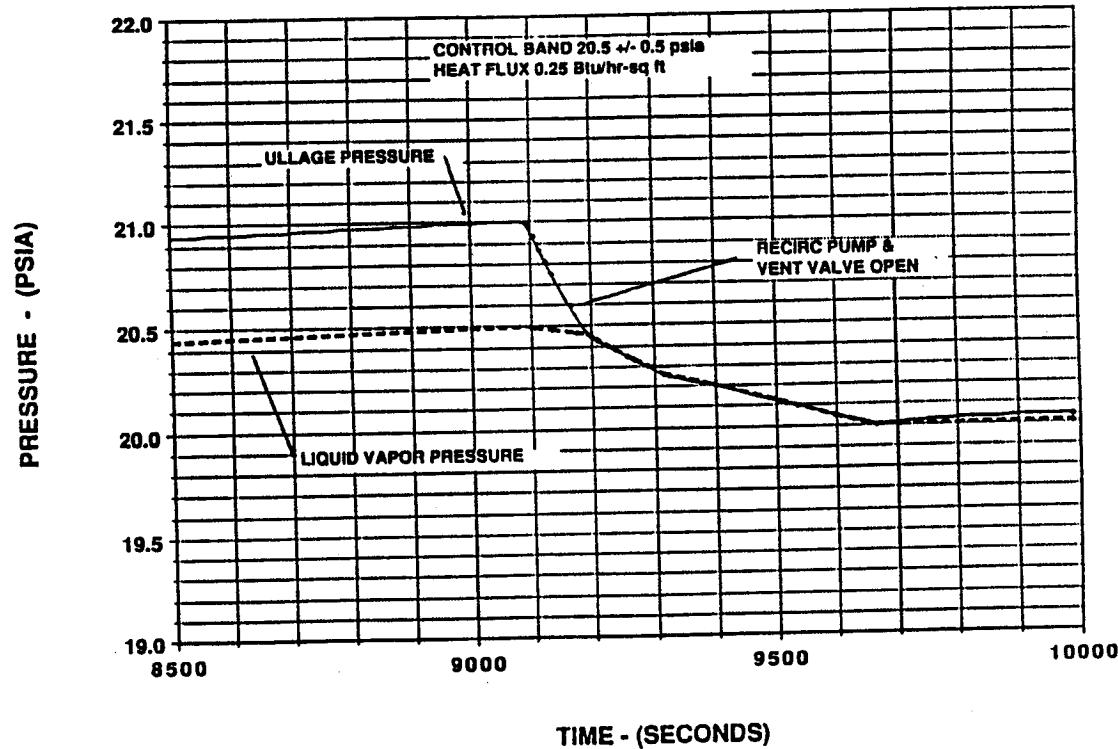


Figure 2.6.4 TVS Performance Simulation at 25% Liquid Quantity

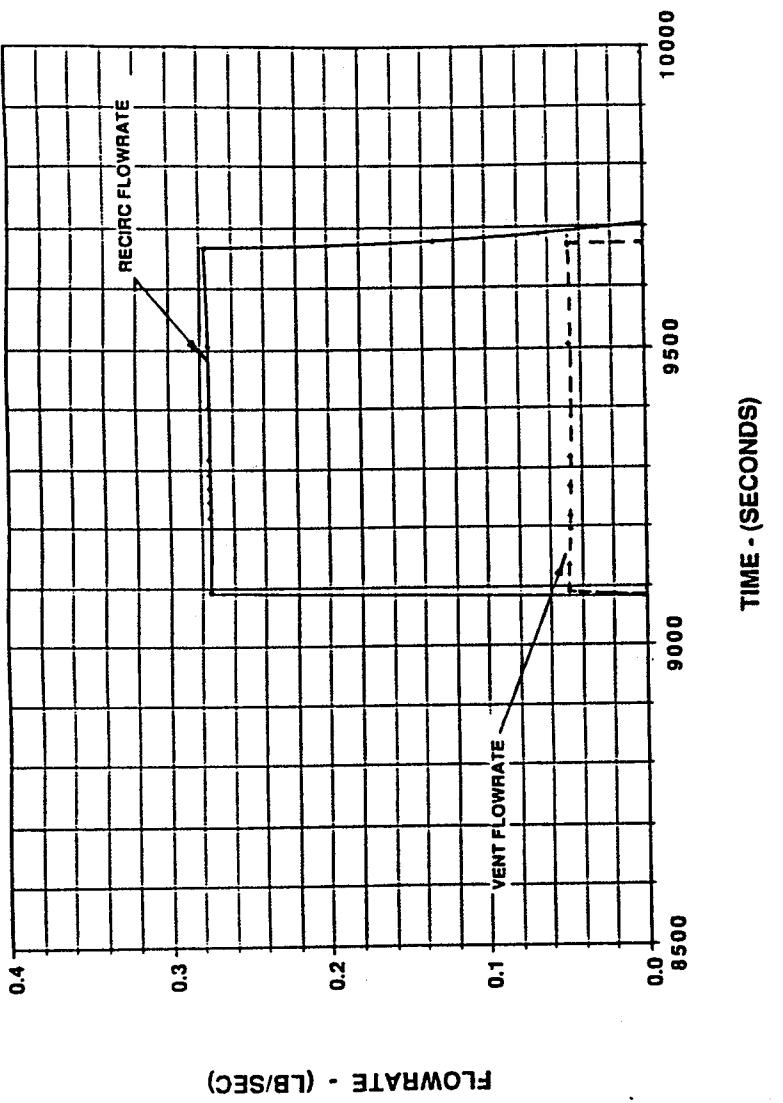


Figure 2.6.5 TVS Recirculation Pump and Vent Valve Flow Rate Transient During Ullage Destratification (25% Liquid Quantity)

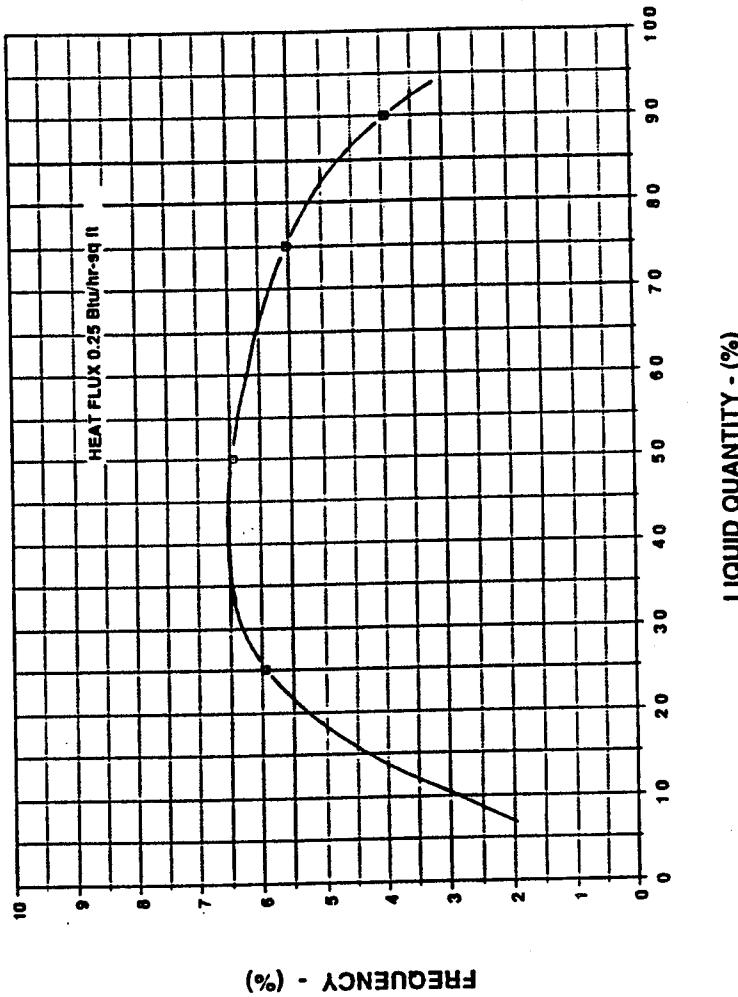


Figure 2.6.6 TVS Operation Frequency (Percent) as a Function of Liquid Quantity

SECTION 3

COMPUTER MODEL DESCRIPTION

3.1 Programming Description

The Zero-g TVS performance prediction program was developed on the following system

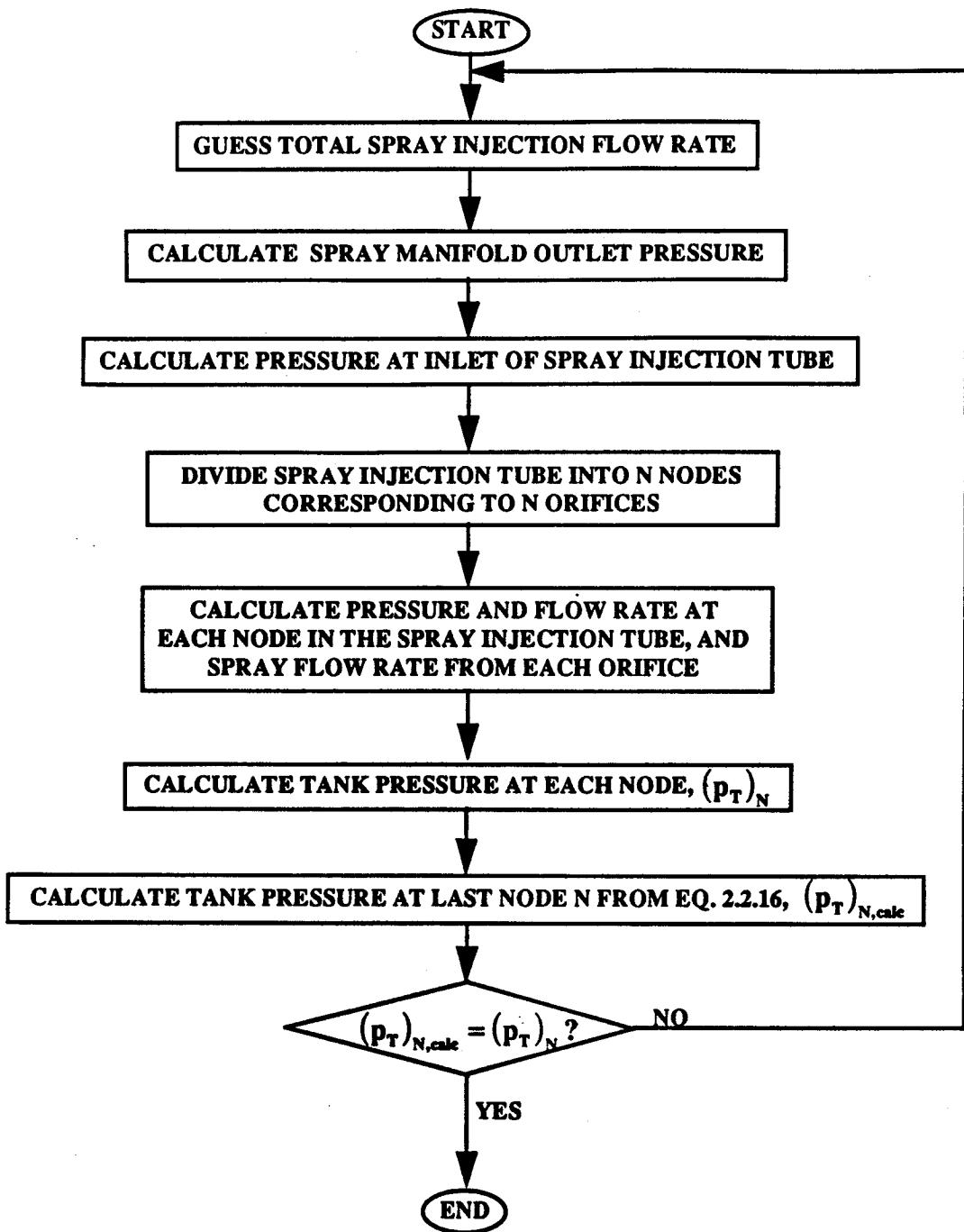
Computer	HP-9000 Series 500
Operating System	HP-UX rel. 5.2.1
Language	FORTRAN 77 rel. 5.12
Plotter	HP-7550
Plotting Software	CRTPLT

3.2 Flow Charts

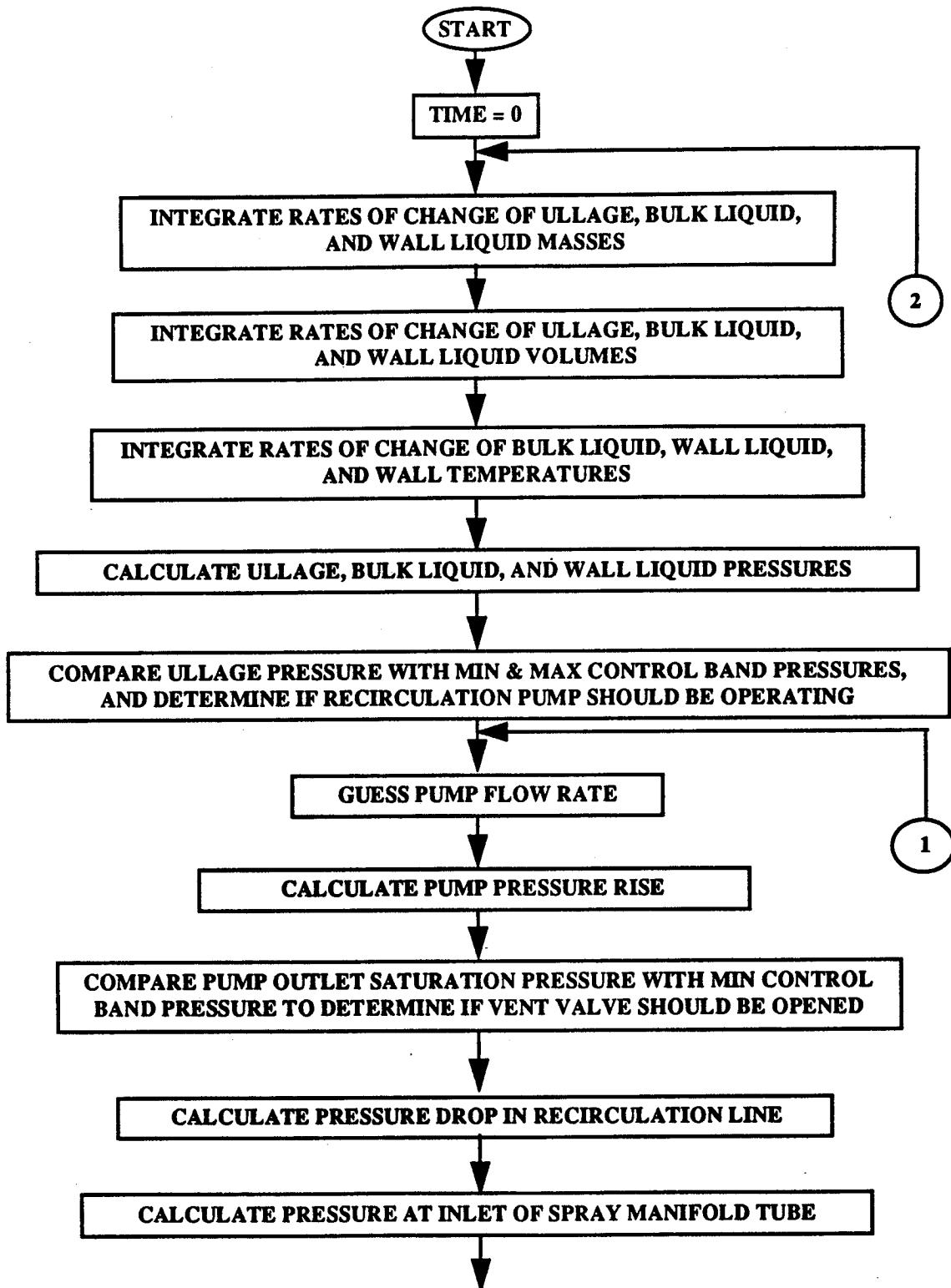
3.2.1 Heat Exchanger Model

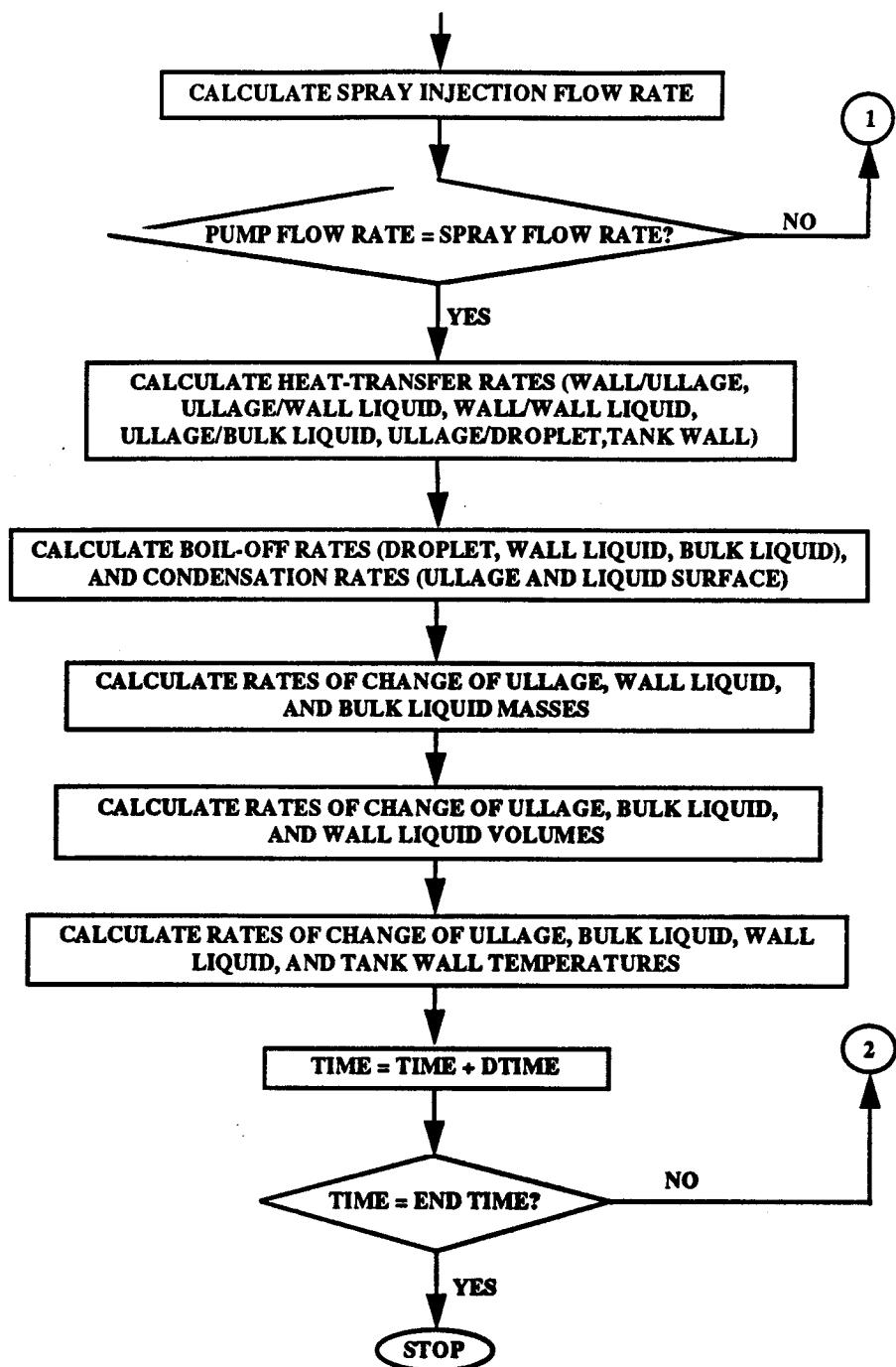
The flow chart of the heat exchanger model is shown in Section 1.4 of Reference 29.

3.2.2 Spray Manifold/Injection Tube Model



3.2.3 Integrated Zero-g TVS Model





3.3 Definition of Variables

3.3.1 Input Variables

3.3.1.1 Heat Exchanger Model

The input variables of the heat exchanger model are described in Section 3.2.1 of Reference 29.

3.3.1.2 Spray Manifold/Injection Tube Model

<u>SYMBOL</u>	<u>UNIT</u>	<u>DESCRIPTION</u>
dsm	in	spray manifold diameter
zsm	in	spray manifold length
dsi	in	spray injection tube diameter
zsi	in	spray injection tube length
norf		number of orifices
nbar		number of spray bars
ks		spray orifice loss coefficient
cds		spray orifice discharge coefficient
roverd		bend r/d of spray manifold
dmdot	lb _m /sec	flow rate increment
tol	psia	convergence tolerance on tank pressure
nlim		max number of iterations
nsec		number of sections with the same orifice sizes
node		node number
dorf	in	orifice diameter

3.3.1.3 Recirculation Pump Model

<u>SYMBOL</u>	<u>UNIT</u>	<u>DESCRIPTION</u>
----------------------	--------------------	---------------------------

mdotd	lb _m /sec	design pump flow rate
dpd	psid	design pump pressure rise
npumpi	rpm	initial pump speed
npumpd	rpm	design pump speed
xhp		multiplier to determine design input horsepower
xn		fraction of design pump speed used to determine the pump speed operating band
deltat	sec	time needed to reach design speed
effp		pump efficiency

3.3.1.4 Integrated Zero-g TVS Model

<u>SYMBOL</u>	<u>UNIT</u>	<u>DESCRIPTION</u>
xd		multiplier for spray injection orifice size (to determine droplet size)
xchar	in	characteristic length used in the equation to calculate the number of droplets
he		helium injection indicator (1 = yes, 0 = no)
xcond		multiplier for liquid surface condensation rate
prtsp		time indicator to print output of subroutine spray (output is printed prtsp<time<prtsp+0.1)
pui	psia	initial ullage pressure
tui	R	initial ullage temperature
pli	psia	initial ullage temperature
twi	R	initial wall temperature
twli	R	initial wall liquid temperature
full	%	percent full level

x11	in	length of straight section downstream of recirculation line 90-degree bend
dtank	in	tank diameter
hcyl	in	cylinder height
hbulk	in	tank bulkhead height
tkw	in	tank wall thickness
dsb	in	spray bar diameter
d1	in	diameter of recirculation line upstream of reducer
d2	in	diameter of recirculaton line downstream of reducer
mdvent	lb _m /sec	overboard venting flow rate
mdsi	lb _m /sec	initial spray (pump) flow rate
dthex	R	heat exchanger temperature drop
qflux	Btu/hr-ft ²	heat flux
g	ft/sec ²	acceleration level
pmin	psia	control band min pressure
pmax	psia	control band max pressure
delt2	sec	integration time step when pump is off
iprnt2		number of time steps between output printing when pump is off
iplot2		number of time steps between output plotting when pump is off
fintim	sec	end time
delt1	sec	integration time step when pump is on
iprnt1		number of time steps between output printing when pump is on

<i>iplot1</i>		number of time steps between output plotting when pump is on
<i>nline</i>		number of output lines printed per page
<i>ovariable</i>		option to plot <i>variable</i> (1=yes)
<i>subhd</i>		plot subheading
<i>x titl</i>		plot x-title
<i>y variable</i>		y-title of <i>variable</i> plot

3.3.1.5 LH₂ saturation properties

<i>nsat</i>		number of data points
<i>tsat</i>	R	saturation temperature
<i>psat</i>	psia	saturation pressure
<i>enthf</i>	Btu/lb _m	saturated liquid enthalpy
<i>shpf</i>	Btu/lb _m -R	saturated liquid specific heat
<i>densf</i>	lb _m /ft ³	saturated liquid density
<i>texpf</i>	R ⁻¹	saturated liquid thermal expansion coefficient
<i>condf</i>	Btu/hr-ft-R	saturated liquid thermal conductivity
<i>viscf</i>	lb _m /ft-sec	saturated liquid dynamic viscosity
<i>enthg</i>	Btu/lb _m	saturated vapor enthalpy

3.3.1.6 GH₂ Properties

<i>np (nt)</i>		number of pressures (temperatures)
<i>t nrm</i>		normalized temperature
<i>t const</i>	R	reference temperature
<i>p vap</i>	psia	pressure
<i>t vap</i>	R	temperature

enth	Btu/lb _m	enthalpy
shv	Btu/lb _m -R	specific heat at constant vapor
shp	Btu/lb _m -R	specific heat at constant pressure
dens	lb _m /ft ³	density
cond	Btu/hr-ft-R	thermal conductivity
visc	lb _m /ft-sec	dynamic viscosity

3.3.2 Output Variables

3.3.2.1 Heat Exchanger Model

The output variables of the heat exchanger model are described in Section 3.2.2 of Reference 29.

3.3.2.2 Spray Manifold/Injection Tube Model

<u>SYMBOL</u>	<u>UNIT</u>	<u>DESCRIPTION</u>
mdot	lb _m /sec	spray (pump) flow rate
dppump	psid	pump pressure rise
dpsm	psi	spray manifold tube pressure drop
dpsi	psi	spray injection tube pressure drop
pin	psia	nodal inlet pressure
pout	psia	nodal outlet pressure
pnode	psia	nodal pressure
ptank	psia	tank pressure
mdin	lb _m /sec	nodal inlet mass flow rate
mdout	lb _m /sec	nodal outlet mass flow rate
rds	lb _m /sec	spray flow rate through each orifice

veld	ft/sec	droplet velocity
as	in ²	spray orifice area

3.3.2.3 Tank Model

<u>SYMBOL</u>	<u>UNIT</u>	<u>DESCRIPTION</u>
pu	psia	ullage pressure
tu	R	ullage temperature
vu	ft ³	ullage volume
mu	lb _m	ullage mass
tw	R	wall temperature
pl	psia	bulk liquid pressure
tl	R	bulk liquid temperature
vl	ft ³	bulk liquid volume
ml	lb _m	bulk liquid mass
pwl	psia	wall liquid pressure
twl	R	wall liquid temperature
vwl	ft ³	wall liquid volume
mw1	lb _m	wall liquid mass
mv	lb _m	mass vented overboard
ts	R	spray temperature
mdlu	lb _m /sec	bulk liquid boil-off rate
mds	lb _m /sec	spray (pump) flow rate
mdsl	lb _m /sec	spray flow rate into bulk liquid
mdsu	lb _m /sec	spray flow rate into ullage

mddu	lb _m /sec	droplet evaporation rate
mdbw	lb _m /sec	wall liquid boil-off rate
mdul	lb _m /sec	ullage condensation rate
mdcond	lb _m /sec	liquid surface condensation rate
qwu	Btu/sec	heat-transfer rate between wall and ullage
quwl	Btu/sec	heat-transfer rate between ullage and wall liquid
qwl	Btu/sec	heat-transfer rate between wall and wall liquid
qul	Btu/sec	heat-transfer rate between ullage and bulk liquid
qud	Btu/sec	heat-transfer rate between ullage and droplet
qus	Btu/sec	heat-transfer rate between ullage and (unsubmerged) spray bar
qls	Btu/sec	heat-transfer rate between bulk liquid and (submerged) spray bar
npump	rpm	pump speed
dppump	psid	pump pressure rise

3.4 Program Listing

3.4.1 Heat Exchanger Model

*

* Zero G Venting System Integrated Steady State Heat Exchanger
* Performance Program

*

* By Tibor Lak, David Soo Hoo, & Dr. Han Nguyen

*

* A HP-9000 Program adapted from the Shuttle Venting Program, Rev. 3
* Includes single & two phase flow heat transfer & pressure losses.

*

* May 8, 1992

*

*** INITIAL CONDITIONS:**

*

* MASSIC - MASS IN THE TANK OR MANIFOLD TO BE VENTED (LBM)
* PIC - SATURATED PRESSURE OF THE TANK OR MANIFOLD (PSIA)
* PSTI - INITIAL GUESS AT THE INLET PRESSURE DURING BOILING (PSIA)
* MDOT - INITAL GUESS AT THE VENT OR LEAK FLOWRATE (LB/SEC)

*

*** FLUID PROPERTIES:**

*

* Prop - Type of Propellant (1.0 = Hydrogen, 2.0 = Oxygen)
* PTP - TRIPLE POINT PRESSURE (PSIA)
* AVLIG - SONIC VELOCITY IN LIQUID (FT/SEC)

*

*** EXTERNAL CONDITIONS:**

*

* PAMB - AMBIENT PRESSURE (PSIA)
* G - GRAVITY

*

*** INTERNAL CONDITIONS & CONFIGURATIONS:**

*

* M - NUMBER OF NODES IN THE VENT PATH (Maxium is 20 nodes)
* VT - TANK OR MANIFOLD VOLUME (CUBIC IN)
* AX - EXIT AREA (SQ IN)
* A(1-M) - FLOW AREA OF THE VENT PATH PER NODE (SQ IN)
* K(1-M) - FLOW LOSS COEFFICIENT PER NODE
* DH(1-M) - CHANGE IN HEIGHT BETWEEN NODES (IN)
* QDOT - HEAT FLUX INTO THE FLUID PER NODE (BTU/SEC)
* (1-M)
* SAREA - SURFACE AREA OF THE NODE: USED TO CALCULATE THE
* (1-M) HEAT TRANSFER COEFFICIENT

- * LENGTH - LENGTH OF NODES (IN)
- * (1-M)
- *
- * VARIABLE EXIT AREA VARIABLES:
- *
- * VA - Change in area (1 = yes , 0 = no)
- * G1 - Acceleration in number of gravities.
- * DIAM - Line diameter (in)
- * SMAX - Maximum exposed surface area (sq inches)
- *
- * PROGRAM CONTROL VARIABLES:
- *
- * FINTIM - MAXIMUM RUN TIME (SEC)
- * PRDEL - PRINT INTERVAL (SEC)
- * QDTERR - QDOT ITERATION ERROR
- * Delptp - Delta Exit Pressure From the Triple Point Pressure
- * DELT - TIME INTERVAL BETWEEN ITERATIONS (SEC)
- * OT - option to plot time vs various parameters (0 = no plot)
- * OM - option to plot mass vs various parameters (0 = no plot)
- * OP - option to plot pressure vs various parameters (0 = no plot)
- * DEBUG - option to print different information between iterations
- * [0.0 = no debug] [1.0 = results from mdot loop]
- * [2.0 = results from PST loop] [3.0 = results from PES loop]
- * [4.0 = results from PSFO loop] [5.0 = results from PSFI loop]
- * [6.0 = results from Liq PSFO loop] [7.0 = results from PXS loop]
- * [8.0 = results from gas calc. loops]
- * [9.0 = results from all 1 thru 7 loops]
- *
- double precision pic,psti,ptp,avliq,pamb,vt,errmx,permx
- double precision delt,delptp,ys,sic,time,rhot,tsp,tts
- double precision pst,pst1,slt,sgt,rholt,rhogt,yt,rhotv,ptsc,ts1
- double precision berr,ps,pliq,liq,hls,hgs,dmdot,pl,ph
- double precision rhole,rhoge,hle,hge,sle,sge,he,qdterr
- double precision rhoflo,rhogfo,hlf,hgfo,slfo,sgfo,hfo,rhofo,pdfo
- double precision htot,pfo,psfo,visgfo,vislfo,asvo,drhd1,to,qerr
- double precision rholfi,rhogfi,hlf,hgfi,slfi,sgfi,hfi,rhofi,pdfi
- double precision xf,psfi,visgfi,vislfi,asvi,lgxf,phisqf,dpff,psfn
- double precision dptf,dpmf,perr,del,pxs,hlx,hgx,slx,sgx,rholx,yx
- double precision hx,vx,rhox,rhoxv,pxsc,gerr,tx,pxd,px,thrust
- double precision isp,dpx,sfo1,hlf,slf,psx1,fm,fp,rhogx
- double precision tdim,ti,tw,pback
- *
- double precision pes(2),tes(2),htote(2),pde(2),rhoe(2),hs(2)
- *
- double precision RHOLI(45,2),RHOLO(45,2),RHOGI(45,2),RHOGO(45,2)

```

double precision PSO(45,2),PDI(45,2),PDO(45,2),HI(45,2),HO(45,2),
1           TSI(45,2),tave(45,2)
double precision RHOO(45,2),SI(45,2),SO(45,2),pi(45,2),po(45,2),
1           tso(45,2)
double precision VI(45,2),VO(45,2),MACHI(45,2),MACHO(45,2)
double precision DPF(45,2),PHISQ(45,2),DPT(45,2),DPM(45,2),
1           ERR(45,2),rhoi(45,2)
double precision XTT(45,2),QDTT(45,2),RN(45,2),psi(45,2)
double precision x1(50),p2(50),p3(50),p4(50),p5(50),p6(50),p7(15)
double precision p8(50),p9(50),p10(50),lmxpar(50),lvisp(50)
double precision lrhop(50),grhop(50),rhovp(50),lent(50),lentp(50)
double precision gent(50),sclent(15,15),p11(50),p12(50),drhdto(50)
double precision t7(15,15),gvisp(50),gentp(50)
double precision cpg(6,14),thkg(6,14),t14(14),viscg(6,14),
1           p14(6),delta(45)
real yo(45,2),yi(45,2),a(45,2),k(45,2),qdot(45,2),dh(45,2)
real mass,kf,massic,mlow,mhigh,mdot(2),twall(45),ye(2),mdt,m8
character *8 name(10),subt(10)
integer exit,ot,om,op,num(15),va
*
* Define Heat Transfer Variables
*
real length(45),sarea(45,2)
double precision p13(50),surft(50),q(45,2),cpl(50),thkl(50),
1           stemp(50),cp,ts,st,cond
*
* Define Plot Variables
*
real pttitle(18,4)
real subhd(18),name1(10),name2(10),name3(10),name4(10)
real name6(10),name7(10),name8(10),name9(10),name10(10)
real name11(10),name12(10),name5(10)
common /pltcom/ misc(3),nc,miss(13),dclim,ltick,nfig,npmin,
1           nlines,nchlin,pttitle
c
data name/'mdot ','isp ','thrust ','qerr '
1     , 'rhot ','pst ','pes ','htot '
2     , 'ext area','twall '
*
* Read Input Data
*
read (5,100) lable
read (5,101) massic,pic,psti,mdot(1),pbck
read (5,100) lable
read (5,101) prop,ptp,avliq,pamb,g

```

```

read (5,100) lable
read (5,102) m,vt,ax,va,tw
read (5,100) lable
read (5,*) mdot(2),disp,dth,vtdi,ed
read (5,100) lable
read (5,101) dpinc,errmx,perrmx,debug,exdi
read (5,100) lable
read (5,101) fintim,prdel,qdterr,delt,delptp
read (5,100) lable
read (5,104) ot,om,op,dq
read (5,100) lable
do i=1,m
    read (5,105) a(i,1),dh(i,1),qdot(i,1),length(i)
enddo
read (5,106) subhd
read (5,107) subt
100 format (//,a1)
101 format (5f10.0)
102 format (i10,2f10.0,i10,f10.0)
103 format (4f10.0,e10.1)
104 format (3i10,f10.0)
105 format (4f10.0)
106 format (/18a4)
107 format (10a8)
*
*      Write Input Data
*
if (debug .eq. 1.0 .or. debug .ge. 10.0) then
    write (6,101) massic,pic,psti,mdot(1),pback
    write (6,101) prop,ptp,avliq,pamb,g
    write (6,102) m,vt,ax,va,tw
    write (6,*) mdot(2),disp,dth,vtdi,ed
    write (6,101) dpinc,errmx,perrmx,debug,exdi
    write (6,101) fintim,prdel,qdterr,delt,delptp
    write (6,104) ot,om,op,dq
    do i=1,m
        write (6,105) a(i,1),dh(i,1),qdot(i,1),length(i)
    enddo
endif
*
*      read tables of data file misc.data
*
if (prop.eq.1.0) then
    open (unit=2, file='h2misc.data',status='unknown')
else

```

```

open (unit=2, file='o2misc.data',status='unknown')
endif
read (2,150) n1
do i=1,n1
    read (2,155) x1(i),lmxpar(i)
enddo
*
read (2,150) n2
do i=1,n2
    read (2,155) p2(i),lvisp(i)
enddo
*
read (2,150) n3
do i=1,n3
    read (2,155) p3(i),gvisp(i)
enddo
*
if (prop .ne. 1.0) then
    read (2,150) n12
    do i=1,n12
        read (2,155) p12(i),drhdto(i)
    enddo
endif
*
*   read tables of data file rho.data
*
if (prop.eq.1.0) then
    open (unit=3, file='h2rho.data',status='unknown')
else
    open (unit=3, file='o2rho.data',status='unknown')
endif
read (3,150) n4
do i=1,n4
    read (3,155) p4(i),lrhop(i)
enddo
*
read (3,150) n5
do i=1,n5
    read (3,155) p5(i),grhop(i)
enddo
*
read (3,150) n6
do i=1,n6
    read (3,155) rhovp(i),p6(i)
enddo

```

```

*
read (3,151) n7,maxt
do i=1,n7
  read (3,*) p7(i),num(i)
  do j=1,num(i)
    read (3,155) t7(i,j),sclent(i,j)
  enddo
enddo
*
*   read tables of data file enthalpy & entropy data
*
if (prop.eq.1.0) then
  open (unit=4, file='h2ent.data',status='unknown')
else
  open (unit=4, file='o2ent.data',status='unknown')
endif
read (4,150) n8
do i=1,n8
  read (4,155) p8(i),lentp(i)
enddo
*
read (4,150) n9
do i=1,n9
  read (4,155) p9(i),gentp(i)
enddo
*
read (4,150) n10
do i=1,n10
  read (4,155) p10(i),lent(i)
enddo
*
read (4,150) n11
do i=1,n11
  read (4,155) p11(i),gent(i)
enddo
*
*   read tables of data file thermal conductivity & surface tension data
*
if (prop.eq.1.0) then
  open (unit=10, file='h2thermo.data',status='unknown')
*
read (10,150) n13
do i=1,n13
  read (10,*) p13(i),cpl(i),thkl(i),surft(i),stemp(i)
enddo

```

```

c      pst = pst + beta/2.0
c      else
c          pst = pst - beta/2.0
c      endif
c          j1 = j1 + 1
c          if (j1.gt.100) go to 6
c          go to 5
c      endif
c
6 PS=PST
*
IF (debug .eq. 2.0 .or. debug .ge. 10.0) THEN
    write (6,110) j1,pst,slt,sgt,rholt,rhogt,ptsc,mass,yt
110  format ('counter = ',i4,4x,'PST = ',f8.3,/,7(3x,f8.3))
    ENDIF
*
PSTI=PST
PLIQ=PST
if (prop.eq.1.0) then
    call H2SAT(PST,tliq)
else
    call o2sat(PST,tliq)
endif
YS=YT
*
* SATURATED LIQUID AT SOURCE
*
IF(PLIQ.le.PST) then
    call value(n8,p8,lentp,PST,hls)
    call value(n9,p9,gentp,PST,hgs)
    HS(1) = (1.0-YS)*HLS+YS*HGS
        HS(2) = HS(1)
else
*
* SUBCOOLED LIQUID AT SOURCE
*
YS=0.0
call value3(np7,maxt,num,p7,t7,sclent,pliq,tliq,hls)
HS(1) = HLS
    HS(2) = HS(1)
endif
*
* Set Constants for Mdot Calculation
*
tsp = PTP

```

c START ITERATION ON THE FLOWRATE CONVERGENCE BASED ON CHOKE
FLOW |
c AT THE ORIFICE BY THE OUTLET OF THE TANK FOR THE TVS SYSTEM |

c-----

THETA = 1.05

DO L1 = 1, 1000

if (tagl .ne. 0.0 .and. tagh .ne. 0.0) then

mdot(1) = mlow + dm * (pl / (pl - ph))

else

mdot(1) = (theta + 1.0) * mdot(1)/2.0

endif

*

* DEFINITION OF TANK EXIT CONDITION (SONIC/SUBSONIC FLOW)

* (CHOKE FLOW AT THE TANK EXIT NODE M)

*

HTOTE(1)=HS(1)+QDOT(M,1)/MDOT(1)-G*DH(M,1)/(12.0*1728.0)

AT = A(M,1)

seold = 0.0

xpes = 0.9

tsp = xpes * tsp

do i = 1,400

tsp = tsp + dpinc

IF(tsp.GT.PST) GO TO 15

IF(tsp.LT.PTP) tsp = ptp

call value(n4,p4,lrhop,tsp,rhole)

call value(n5,p5,grhop,tsp,rhog)

call value(n8,p8,lentp,tsp,hle)

call value(n9,p9,gentp,tsp,hge)

call value(n10,p10,lent,tsp,sle)

call value(n11,p11,gent,tsp,sge)

call TPHS(HTOTE(1),RHOE,RHOG,MDOT(1),AT,HLE,HGE,SLE,

1 sge,ye(1),he,se,rhoec(1),pde(1))

if (se .le. seold) goto 15

seold = se

enddo

15 continue

if (prop.eq.1.0) then

call H2SAT(tsp,tts)

else

call o2sat(tsp,tts)

endif

*

*

PO(M,1)=tsp+PDE(1)

YO(M,1)=YE(1)

```

*
* PRESSURE LOSS CALCULATION THROUGH NODE M
*
    KF=K(M,1)
    AREA=A(M,1)
*
* CHANGE IN ELEVATION
*
    DHI=DH(M,1)
    DHO=DH(M+1,1)
    HO(M,1)=HS(1)+QDOT(M,1)/MDOT(1)-G*DH(M,1)/(12.0*1728.0)
    HTOT=HO(M,1)
*
* TWO-PHASE FLOW REGION
*
    PFO=PO(M,1)
    PSFO=PFO
*
* DETERMINE THE OUTLET CONDITIONS AT NODE N, GIVEN THE TOTAL
* PRESSURE AT THE OUTLET
*
    DO I1 = 1,5
        IF(PSFO.LT.PTP) PSFO=PTP
        IF(PSFO.GT.PST) PSFO=PST
        call value(n4,p4,lrhop,PSFO,rholfo)
        call value(n5,p5,grhop,PSFO,rhogfo)
        call value(n8,p8,lentp,PSFO,hlf0)
        call value(n9,p9,gentp,PSFO,hgfo)
        call value(n10,p10,lent,PSFO,slfo)
        call value(n11,p11,gent,PSFO,sgfo)
        call value(n3,p3,gvisp,PSFO,visgfo)
        call value(n2,p2,lvisp,PSFO,vislfo)
        call TPHS(HTOT,RHOLFO,RHOGFO,MDOT(1),AREA,HLFO,HGFO,SLFO,
1           sgfo,yfo,hfo,sfo,rhofo,pdfo)
        PSFO=PFO-PDFO
*
* End of I1 Loop (To Determine Node M Outlet Conditions)
*
    enddo
    if (prop.eq.1.0) then
        call H2SAT(PSfo,ts1)
    else
        call o2sat(PSfo,ts1)
    endif
*

```

```

PSO(M,1)=PSFO
PDO(M,1)=PDFO
RHOL0(M,1)=RHOLFO
RHOG0(M,1)=RHOGFO
RHOO(M,1)=RHOFO
YO(M,1)=YFO
HO(M,1)=HFO
SO(M,1)=SFO
VO(M,1)=144.0*MDOT(1)/(A(M,1)*RHOO(M,1))
TSO(M,1)=ts1
if (prop.le.1.0) then
  call H2SVEL(YFO,PSFO,RHOLFO,RHOGFO,asvo)
else
  call value(n12,p12,drhdto,psfo,drhd1)
  call o2sat(psfo,to)
  call o2svel(yfo,to,rholfo,rhogfo,drhd1,asvo)
endif
MACHO(M,1)=VO(M,1)/ASVO
IF(MACHO(M,1).LT.0.0) MACHO(M,1)=0.0
IF(MACHO(M,1).GT.1.0) MACHO(M,1)=1.0
*
* INITIAL GUESS AT NODE INLET BASED ON COMPRESSIBLE LOSS
*
* call INITL(PSFO,KF,MDOT(1),AREA,RHOLFO,psfi)
*
* TOTAL ENTHALPY AT NODE INLET
*
* HI(M,1)=HS(1)+QDOT(M+1,1)/MDOT(1)-G*D(H,M,1)/(12.0*1728.0)
* HTOT=HI(M,1)
*
* DEFINITION OF STATIC PRESSURE AT NODE INLET
*
DO I2 = 1,50
  IF(PSFLLT.PTP) PSFI=PTP
  IF(G.LE.0.0.AND.PSFLLT.PSFO) PSFI=PSFO
  IF(PSFLGT.PST) PSFI=PST
  call value(n4,p4,lrh0,PSFI,rholfi)
  call value(n5,p5,grh0,PSFI,rhogfi)
  call value(n8,p8,lentp,PSFI,hlf1)
  call value(n9,p9,gentp,PSFI,hgfi)
  call value(n10,p10,lent,PSFI,slfi)
  call value(n11,p11,gent,PSFI,sgfi)
  call value(n3,p3,gvisp,PSFI,visgfi)
  call value(n2,p2,lvisp,PSFI,vislfi)
  call TPHS(HTOT,RHOLFI,RHOGFI,MDOT(1),AREA,HLFI,HGFI,SLFI,

```

```

1      sgfi,yfi,hfi,sfi,rhofi,pdfi)
PFI=PSFI+PDFI
*
*   LOCKHART-MARTINELLI TWO-PHASE FLOW PARAMETER
*
*   call XPARAM(YFO,VISLFO,VISGFO,RHOLFO,RHOGFO,YFI,VISLFI,
1      visgfi,RHOLFI,RHOGFI,xf)
IF(XF.LT.0.01) XF=0.01
LGXF=LOG10(XF)
call value(n1,x1,lmxpath,LGXF,phisqf)
PHISQF=(10.0**PHISQF)**2
*
*   TWO-PHASE FLOW PRESSURE LOSS AND NODE INLET PRESSURE (PSFN)
*
*   call TPS(AREA,G,DHI,DHO,KF,PSFO,YFO,RHOLFO,RHOGFO,YFI,
1      rholfi,RHOGFI,MDOT(1),PHISQF,dpff,psfn,dptf,
2      dpmf)
ERR(M,1)=ABS(PSFN-PSFI)
PSFI=(PSFI+PSFN)/2.0
*
*   CONDITION WHERE THE INLET STATIC PRESSURE OF THE NODE IS
DETERMINED
*
*   IF(ERR(M,1).LE.ERRMX) GO TO 20
*
*   End of I2 Loop (To Determine Node N Inlet Conditions)
*
enddo
20 PFI=PSFI+PDFI
*
*   CALCULATE THE INLET CONDITIONS OF THE NODE M
*
RHOLI(M,1)=RHOLFI
RHOGI(M,1)=RHOGFI
RHOI(M,1)=RHOIFI
DPM(M,1)=DPMF
DPF(M,1)=DPFF
DPT(M,1)=DPTF
PI(M,1)=PFI
PDI(M,1)=PDFI
PSI(M,1)=PSFI
YI(M,1)=YFI
HI(M,1)=HFI
SI(M,1)=SFI
PHISQ(M,1)=PHISQF

```

```

VI(M,1)=144.0*MDOT(1)/(A(M,1)*RHOI(M,1))
  if (prop.le.1.0) then
    call H2SVEL(YFI,PSFI,RHOLFI,RHOGFI,asvi)
  else
    call value(n12,p12,drhdto,psfi,drhd2)
    call o2sat(psfi,ti)
    call o2svel(yfi,ti,rholfi,rhogfi,drhd2,asvi)
  endif
  MACHI(M,1)=VI(M,1)/ASVI
  IF(MACHI(M,1).LT.0.0) MACHI(M,1)=0.0
  IF(MACHI(M,1).GT.1.0) MACHI(M,1)=1.0
*
* CALCULATE THE TWO PHASE HEAT TRANSFER COEFFICIENT FOR NODE M
* BASED ON INLET PRESSURE
*
  call value(n13,p13,cpl,psfi,cp)
  call value(n13,p13,thkl,psfi,cond)
  call value(n13,p13,surft,psfi,st)
  call value(n13,p13,stemp,psfi,ts)
  if (tave(m,1) .le. 0.0) then
    tave(m,1) = ts
  endif
  hgl = hgfi - hlf1
*
  dtemp = twall(m) - tave(m,1)
*
  di = exdi
*
  dpress = 778.2 * dtemp * rhogfi * hgl / ts
  if (yfi .gt. 0.7) then
    re = 48.0 * mdot(1) * (1.0-0.7)/(pyi * vislfi * di)
  else
    re = 48.0 * mdot(1) * (1.0-yfi)/(pyi * vislfi * di)
  endif
*
  call htcoeff(phase,dtemp,dpress,cp,cond,vislfi,rholfi,
1      rhogfi,st,hgl,re,xf,di,qdott)
  if (yfi .ge. 0.7) then
    tdim = 0.0
    call value2(n14,num1,p14,t14,cpg,psfi,tdim,cp)
    call value2(n14,num1,p14,t14,thkg,psfi,tdim,cond)
    call value2(n14,num1,p14,t14,viscg,psfi,tdim,visgfi)
    re = 48.0 * mdot(1) /(pyi * visgfi * di)
    call htcoeff(3.0,dtemp,dpress,cp,cond,visgfi,rholfi,
1      rhogfi,st,hgl,re,xf,di,qtt)

```

```

        qdott = qdott*(1-yfi) + qtt*yfi
        endif
        q(m,1) = qdott * (twall(m) - tave(m,1)) * sarea(m,1)/144.0
*
        TSI(M,1) = TS
        XTT(M,1) = XF
        RN(M,1) = RE
        QDTT(M,1) = QDOTT
        TAVE(M,1) = (tsi(m,1) + tso(m,1))/2.0
*
        theta = ps/pi(m,1)
        perr = ps-pi(m,1)
        if (perr .lt. 0.0) then
            del = mlow - mdot(1)
            tagl = 1.0
            mlow = mdot(1)
            pl = perr
        else
            del = mhigh - mdot(1)
            tagh = 1.0
            mhigh = mdot(1)
            ph = perr
        endif
        dm = mhigh - mlow
c
        if (dabs(perr) .le. permix .or. dabs(del) .le. 0.0001) goto 30
        enddo
c-----
c    END OF MDOT CALCULATION LOOP
c-----
c
        30 continue
c
c    Determine the Back Pressure to the Heat Exchanger part of the Spray
c    Bar
c
        ptk = psti
        mdt = mdot(2)
        dspry = disp
        call spray(mdt,ptk,dspry,ed,pback)
*****
*   START ITERATION ON THE HEAT TRANSFER COEFFICIENT *
*****
        DO L=1,150
        if (l .gt. 1) then
            do ii = 1,m1

```

```

nn = m - ii
qdot(nn,1) = qdot(nn+1,1) + q(nn,1)
qdot(nn,2) = qdot(nn+1,2) - q(nn,2)
if (sarea(ii,1) .le. 0.0 .or. qdtt(ii,1) .lt. 0.0) then
  c1 = 1.0
else
  c1 = sarea(ii,2) * qdtt(ii,2)/(sarea(ii,1)*qdtt(ii,1))
endif
if (dabs(delta(ii)) .gt. dq) then
  tw = twall(ii)
  twall(ii) = (c1 * tave(ii,2) + tave(ii,1))/(1+c1)
  twall(ii) = (tw + twall(ii))/2.0
  if(twall(ii) .lt. 24.845) twall(ii) = 24.845
endif
enddo
endif
qdt = 0.0
qdt1 = 0.0

```

C-----
 c START THE VENT & SPRAY NODAL NETWORK FLOW PROPERTIES
 CALCULATION !

C-----
 do il = 1, 2
 *
 * DEFINITION OF EXIT CONDITION (SONIC/SUBSONIC FLOW)
 * (CHOKE FLOW AT THE EXIT NODE)
 *
 HTOTE(il)=HS(il)+QDOT(1,il)/MDOT(il)+G*(DH(M+1,il)-DH(1,il))/
 1 (12.0*1728.0)
 AE = A(1,il)
 seold = 0.0
 xpes = 0.9
 PES(il) = xpes * PES(il)
 do i = 1,500
 if (il .eq. 1) then
 PES(il) = PES(il) + dpinc
 else
 PES(il) = pback
 endif
 IF(PES(il).GT.PST) GO TO 40
 IF(PES(il).LT.PTP) PES(il)=PTP
 call value(n4,p4,lrhop,PES(il),rhole)
 call value(n5,p5,grhop,PES(il),rhoge)
 call value(n8,p8,lentp,PES(il),hle)
 call value(n9,p9,gentp,PES(il),hge)


```

    else
        K(1,1) = 0.5
    endif
else
    dd = disp / 12.0
    call frict(dd,ed,mdot(2),visc,ff)
    K(n,2) = ff * length(n)/disp
endif
KF=K(N,il)
*
* CHANGE IN ELEVATION
*
DHI=DH(N,il)
DHO=DH(N+1,il)
HO(N,il)=HS(il)+QDOT(N,il)/MDOT(il)+G*(DH(M+1,il)-DH(N,il))/(
1      (12.0*1728.0)
HTOT=HO(N,il)
IF(YO(N,il).LE.0.0) PHASE=1.0
IF(YO(N,il).GT.0.0) PHASE=2.0
*
* DETERMINE THE PHASE OF THE FLUID
*
IF(PHASE.ge.2.0) then
*::::::::::::::::::
* TWO-PHASE FLOW REGION
*::::::::::::::::::
PFO=PO(N,il)
PSFO=PFO
*
* DETERMINE THE OUTLET CONDITIONS AT NODE N, GIVEN THE TOTAL
* PRESSURE AT THE OUTLET
*
DO I1 = 1,5
IF(PSFO.LT.PTP) PSFO=PTP
IF(PSFO.GT.PST) PSFO=PST
call value(n4,p4,lrhop,PSFO,rholfo)
call value(n5,p5,grhop,PSFO,rhogfo)
call value(n8,p8,lentp,PSFO,hlf0)
call value(n9,p9,gentp,PSFO,hgfo)
call value(n10,p10,lent,PSFO,slfo)
call value(n11,p11,gent,PSFO,sgfo)
call value(n3,p3,gvisp,PSFO,visgfo)
call value(n2,p2,lvisp,PSFO,vislfo)
call TPHS(HTOT,RHOLFO,RHOGFO,MDOT(il),AREA,HLFO,HGFO,
1           slfo,sgfo,yfo,hfo,sfo,rhofo,pdfo)

```

```

PSFO=PFO-PDFO
*
* End of I1 Loop (To Determine Node N Outlet Conditions)
*
      enddo
*
      IF (debug .eq. 4.0 .or. debug .ge. 10.0) THEN
        write (6,112) n,i1,psfo,slfo,sgfo,rhofo,yfo,sfo,hfo,pdfo
112    format ('node #',i2,4x,'counter = ',i4,4x,'PSFO = ',
1           f8.3/,7(3x,f8.3))
        ENDIF
*
      PSO(N,il)=PSFO
      PDO(N,il)=PDFO
      RHOLO(N,il)=RHOLFO
      RHOGO(N,il)=RHOGFO
      RHOO(N,il)=RHOFO
      YO(N,il)=YFO
      HO(N,il)=HFO
      SO(N,il)=SFO
      VO(N,il)=144.0*MDOT(il)/(A(N,il)*RHOO(N,il))
*
* calculate the outlet temperature at node 1
*
      if (n .eq. 1) then
        call H2SAT(psfo,ts1)
        call value(n13,p13,cpl,psfo,cp)
        hgl = hgfo - hlfo
        qq = qdot(1,il)/mdot(il)
        if (yo(1,il) .gt. 0.99) then
          if (il .eq. 1) then
            tso(1,1) = tsi(m,1) + (qq - hgl)/cp
          else
            tso(1,2) = tliq + (qq + hgl)/cp
          endif
        else
          tso(1,il) = ts1
        endif
      endif
*
      if (prop.le.1.0) then
        call H2SVEL(YFO,PSFO,RHOLFO,RHOGFO,asvo)
      else
        call value(n12,p12,drhdto,psfo,drhd1)
        call o2sat(psfo,to)

```

```

        call o2svel(yfo,to,rholfo,rhogfo,drhdt1,asvo)
    endif
    MACHO(N,il)=VO(N,il)/ASVO
    IF(MACHO(N,il).LT.0.0) MACHO(N,il)=0.0
    IF(MACHO(N,il).GT.1.0) MACHO(N,il)=1.0
*
* INITIAL GUESS AT NODE INLET BASED ON COMPRESSIBLE LOSS
*
*      call INITL(PSFO,KF,MDOT(il),AREA,RHOLFO,psfi)
*
* TOTAL ENTHALPY AT NODE INLET
*
*      HI(N,il)=HS(il)+QDOT(N+1,il)/MDOT(il)+  

1       G*(DH(M,il)-DH(N+1,il))/(12.0*1728.0)  

*      HTOT=HI(N,il)
*
* DEFINITION OF STATIC PRESSURE AT NODE INLET
*
*      DO I2 = 1,50
*          IF(PSFI.LT.PTP) PSFI=PTP
*          IF(G.LE.0.0.AND.PSFI.LT.PSFO) PSFI=PSFO
*          IF(PSFI.GT.PST) PSFI=PST
*          call value(n4,p4,lrhop,PSFI,rholfi)
*          call value(n5,p5,grhop,PSFI,rhogfi)
*          call value(n8,p8,lentp,PSFI,hlf1)
*          call value(n9,p9,gentp,PSFI,hgfi)
*          call value(n10,p10,lent,PSFI,slfi)
*          call value(n11,p11,gent,PSFI,sgfi)
*          call value(n3,p3,gvisp,PSFI,visgfi)
*          call value(n2,p2,lvisp,PSFI,vislfi)
*          call TPHS(HTOT,RHOLFI,RHOGFI,MDOT(il),AREA,HLFI,HGFI,
1           slfi,sgfi,yfi,hfi,sfi,rholfi,pdfi)
*          PFI=PSFI+PDFI
*
* LOCKHART-MARTINELLI TWO-PHASE FLOW PARAMETER
*
*      call XPARAM(YFO,VISLFO,VISGFO,RHOLFO,RHOGFO,YFI,VISLFI,
1           visgfi,RHOLFI,RHOGFI,xf)
*          IF(XF.LT.0.01) XF=0.01
*          LGXF=LOG10(XF)
*          call value(n1,x1,lmxpath,LGXF,phisqf)
*          PHISQF=(10.0**PHISQF)**2
*
* TWO-PHASE FLOW PRESSURE LOSS AND NODE INLET PRESSURE (PSFN)
*

```

```

call TPS(AREA,G,DHI,DHO,KF,PSFO,YFO,RHOLFO,RHOGFO,YFI,
1      rholfi,RHOGFI,MDOT(il),PHISQF,dpff,psfn,dptf,
2      dpmf)
ERR(N,il)=ABS(PSFN-PSFI)
PSFI=(PSFI+PSFN)/2.0
*
* CONDITION WHERE THE INLET STATIC PRESSURE OF THE NODE IS
DETERMINED
*
IF(ERR(N,il).LE.ERRMX) GO TO 50
*
* End of I2 Loop (To Determine Node N Inlet Conditions)
*
enddo
50  PFI=PSFI+PDFI
*
IF (debug .eq. 5.0 .or. debug .ge. 10.0) THEN
write (6,113) n,i2,psfi,slfi,sgfi,rholfi,yfi,sfi,hfi,
1                  phsiqf
113   format ('node #',i2,4x,'counter = ',i4,4x,'PSFI = ',
1           f8.3,/6(3x,f8.3),/3x,f8.3)
      ENDIF
*
* CALCULATE THE INLET CONDITIONS OF THE NODE N
*
RHOLI(N,il)=RHOLFI
RHOGI(N,il)=RHOGFI
RHOI(N,il)=RHOIFI
DPM(N,il)=DPMF
DPF(N,il)=DPFF
DPT(N,il)=DPTF
PI(N,il)=PFI
PDI(N,il)=PDFI
PSI(N,il)=PSFI
YI(N,il)=YFI
HI(N,il)=HFI
SI(N,il)=SFI
PHISQ(N,il)=PHISQF
VI(N,il)=144.0*MDOT(il)/(A(N,il)*RHOI(N,il))
      if (prop.le.1.0) then
      call H2VEL(YFI,PSFI,RHOLFI,RHOGFI,asvi)
      else
      call value(n12,p12,drhdto,psfi,drhd2)
      call o2sat(psfi,ti)
      call o2svel(yfi,ti,rholfi,rhogfi,drhd2,asvi)

```

```

        endif
        MACHI(N,il)=VI(N,il)/ASVI
        IF(MACHI(N,il).LT.0.0) MACHI(N,il)=0.0
        IF(MACHI(N,il).GT.1.0) MACHI(N,il)=1.0
*
* CALCULATE THE TWO PHASE HEAT TRANSFER COEFFICIENT FOR NODE N
* BASED ON INLET PRESSURE
*
        call value(n13,p13,cpl,psfi,cp)
        call value(n13,p13,thkl,psfi,cond)
        call value(n13,p13,surft,psfi,st)
        call value(n13,p13,stemp,psfi,ts)
*
* do nn1 = 1, 2
*
        if (tave(n,il) .le. 0.0) then
            tave(n,il) = ts
        endif
        hgl = hgfi - hlf
*
        if (il .eq. 1) then
            dtemp = twall(n) - tave(n,il)
            di = exdi
        else
            dtemp = tave(n,il) - twall(n)
            di = disp
        endif
*
        dpress = 778.2 * dtemp * rhogfi * hgl / ts
        if (yfi .gt. 0.7) then
            re = 48.0 * mdot(il) * (1.0-0.7)/(pyi * vislfi * di)
        else
            re = 48.0 * mdot(il) * (1.0-yfi)/(pyi * vislfi * di)
        endif
c       write (6,*) dtemp,dpress,cp,cond,vislfi,rholfi,rhogfi,
c      1           st,hgl,re,xf
*
        call htcoeff(phase,dtemp,dpress,cp,cond,vislfi,rholfi,
1           rhogfi,st,hgl,re,xf,di,qdott)
        if (yfi .ge. 0.7) then
            tdim = 0.0
            call value2(n14,num1,p14,t14,cpg,psfi,tdim,cp)
            call value2(n14,num1,p14,t14,thkg,psfi,tdim,cond)
            call value2(n14,num1,p14,t14,viscg,psfi,tdim,visgfi)
            re = 48.0 * mdot(il) /(pyi * visgfi * di)

```

```

call htcoeff(3.0,dtemp,dpress,cp,cond,visgfi,rholfi,
1          rhogfi,st,hgl,re,xf,di,qtt)
      qdott = qdott*(1-yfi) + qtt*yfi
      endif
      if (il .eq. 1) then
      q(n,il) = qdott * (twall(n) - tave(n,il)) * sarea(n,il)
1      /144.0
      else
      q(n,il) = qdott * (tave(n,il) - twall(n)) * sarea(n,il)
1      /144.0
      endif
*
* calculate the inlet temperature at node il
*
      call value(n13,p13,cpl,psfi,cp)
      if (yi(n,il) .gt. 0.99) then
      if (il .eq. 1) then
      tsi(n,il) = tso(n,il) - q(n,il)/(mdot(il) * cp)
      else
      tsi(n,il) = tso(n,il) + q(n,il)/(mdot(il) * cp)
      endif
      else
      tsi(n,il) = ts
      endif
*
      XTT(N,il) = XF
      RN(N,il) = RE
      QDTT(N,il) = QDOTT
      TAVE(N,il) = (tsi(n,il) + tso(n,il))/2.0
*
* enddo
*
      if (debug .ge. 10.0) then
      write (6,120) tsi(n,il),twall(n),tave(n,il),re,qdott,
1                  q(n,il),sarea(n,il)
120      format(4(2x,f14.4),/,3(2x,f14.4))
      endif
      ELSE
*::::::::::
*  SINGLE PHASE FLOW (PHASE=1.0, LIQUID FLOW)
*::::::::::
*  CALCULATION OF OUTLET CONDITIONS GIVEN TOTAL PRESSURE AT NODE
*  OUTLET  (PO(N))
*
      PFO=PO(N,il)

```

```

PSFO=PFO
DO I3 = 1,5
  IF(PSFO.LT.PTP) PSFO=PTP
  IF(PSFO.GT.PST) PSFO=PST
  call value(n4,p4,lrhop,PSFO,rholfo)
  PDFO=144.0*MDOT(il)/A(N,il)*MDOT(il)/A(N,il)/
    (2.0*RHOLFO*32.2)
  PSFO=PFO-PDFO
c
c   End of I3 Loop (To Determine Node N Outlet Conditions)
c
c     enddo
c
IF(PSFO.LT.PTP) PSFO=PTP
IF(PSFO.GT.PST) PSFO=PST
call value(n8,p8,lentp,PSFO,hfo)
call value(n10,p10,lent,PSFO,sfo1)
call value(n13,p13,cpl,psfo,cp)
call value(n2,p2,lvisp,PSFO,vislfo)
*
      if (n .eq. 1) then
        if (il .eq. 1) then
          tso(1,il) = tsi(m,il) + qdot(1,il)/(mdot(il) * cp)
        else
          tso(1,il) = tliq + qdot(1,il)/(mdot(il) * cp)
        endif
      endif
      PSO(N,il)=PSFO
      PDO(N,il)=PDFO
      RHOLO(N,il)=RHOLFO
      RHOGO(N,il)=0.0
      RHOO(N,il)=RHOLFO
      YO(N,il)=0.0
      HO(N,il)=HS(il)+QDOT(N,il)/MDOT(il)+G*
        (DH(M+1,il)-DH(N,il))/(12.0*1728.0)
      SO(N,il)=SFO1
      VO(N,il)=144.0*MDOT(il)/(A(N,il)*RHOO(N,il))
      ASVO=AVLIQ
      MACHO(N,il)=VO(N,il)/ASVO
      IF(MACHO(N,il).LT.0.0) MACHO(N,il)=0.0
      IF(MACHO(N,il).GT.1.0) MACHO(N,il)=1.0
c
      IF (debug .eq. 6.0 .or. debug .ge. 10.0) THEN
        write (6,114) n,i3,psfo,pdfo,tso(n,il),rholfo,hfo,
          so(n,il),vo(n,il),htote(il),tso(n,il)
  1

```

```

114      format ('Node # = ',i4,4x,'Counter = ',i4,4x,
1           'Liquid PSFO = ',f8.3,/,8(3x,f8.3))
      ENDIF
*
*   INLET CONDITION CALCULATIONS
*
      RHOLI(N,il)=RHOLO(N,il)
      RHOGI(N,il)=0.0
      RHOI(N,il)=RHOLI(N,il)
      DPM(N,il)=0.0
      DPF(N,il)=144.0*K(N,il)/(2.0*RHOLI(N,il)*32.2)*
1           (MDOT(il)/A(N,il))**2
      DPT(N,il)=DPF(N,il)
      PI(N,il)= PO(N,il)+DPF(N,il)-G*RHOLI(N,il)*
1           (DH(N,il)-DH(N+1,il))/1728.0
      PDI(N,il)=144.0*(MDOT(il)/A(N,il))**2/
1           (2.0*RHOLI(N,il)*32.2)
      PSI(N,il)=PI(N,il)-PDI(N,il)
      YI(N,il)=0.0
      VI(N,il)=144.0*MDOT(il)/(RHOLI(N,il)*(A(N,il)))
      ASVI=AVLIQ
      MACHI(N,il)=VI(N,il)/ASVI
      PSFI=PSI(N,il)
      IF(PSFI.LT.PTP) PSFI=PTP
      IF(PSFI.GT.PST) PSFI=PST
      call value(n2,p2,lvisp,PSFI,vislfi)
      call value(n8,p8,lentp,PSFI,hlf)
      call value(n10,p10,lent,PSFI,slf)
*
      HI(N,il)=HLF
      HI(N,il)=HS(il)+QDOT(N+1,il)/MDOT(il)+*
1           G*(DH(M,il)-DH(N+1,il))/(12.0*1728.0)
      SI(N,il)=SLF
      PHISQ(N,il)=1.0
c
      IF (debug .eq. 6.0 .or. debug .ge. 10.0) THEN
        write (6,214) n,psfi,pdi(n,il),dpm(n,il),
1           dpf(n,il),dpt(n,il),vi(n,il)
214      format ('Node # = ',i4,4x,'Liquid PSFI = ',f8.3,/,
1           5(3x,f8.3))
      ENDIF
*
*   CALCULATE THE HEAT TRANSFER COEFFICIENT FOR NODE N
*   BASED ON INLET PRESSURE
*
      call value(n13,p13,cpl,psfi,cp)

```

```

call value(n13,p13,thkl,psfi,cond)
call value(n13,p13,surft,psfi,st)
call value(n13,p13,stemp,psfi,ts)
    if (tave(n,il) .le. 0.0) then
        tave(n,il) = tso(n,il)
    endif
    hgl = hgfi - hlfi
*
    if (il .eq. 1) then
        dtemp = twall(n) - tave(n,il)
        di = exdi
    else
        dtemp = tave(n,il) - twall(n)
        di = disp
    endif
*
dpress = 778.2 * dtemp * rhogfi * hgl / ts
re = 48.0 * mdot(il)/(pyi * vislfi * di)
call htcoeff(phase,dtemp,dpress,cp,cond,vislfi,rholfi,
1          rhogfi,st,hgl,re,xf,di,qdott)
if (il .eq. 1) then
q(n,il) = qdott * (twall(n) - tave(n,il)) * sarea(n,il)
1          /144.0
else
q(n,il) = qdott * (tave(n,il) - twall(n)) * sarea(n,il)
1          /144.0
endif
*
if (il .eq. 1) then
    tsi(n,il) = tso(n,il) - q(n,il)/(mdot(il) * cp)
else
    tsi(n,il) = tso(n,il) + q(n,il)/(mdot(il) * cp)
endif
XTT(N,il) = XF
RN(N,il) = RE
QDTT(N,il) = QDOTT
TAVE(N,il) = (tsi(n,il) + tso(n,il))/2.0
*
IF (debug .eq. 6.0 .or. debug .ge. 10.0) THEN
    write (6,314) tsi(n,il),q(n,il),dtemp,xf,re,qdott,
1          hgl,cp,di,tave(n,il)
314    format ('Liquid TSI = ',f8.3,3x,f10.4,2(3x,f8.3),3x,
1          f12.2,/,5(3x,f8.3))
ENDIF
*
```

```

ENDIF
*      PO(N+1,il)=PI(N,il)
c+++++
c   End of N do loop (Conditions for Both the Inlet & Outlet of the   +
c           Nodal Network are Defined)   +
c+++++
c.....+
60    continue
ENDDO
c.....+
c RE-CALCULATE THE NODAL PROPERTIES OF THE NODES THAT ARE 100% GAS
.
c.....+
if (il .eq. 1) then
  do ig = m1, 1, -1
    if (yo(ig,1) .gt. 0.99) goto 130
  enddo
130  phase = 3.0
      dummy = 1.4 * 766.55232 * 32.2
      do ia = ig, 1, -1
        if (yo(ia,1) .ge. 0.99) then
          iia = ia + 1
          psi(ia,1) = pso(iia,1)
          tsi(ia,1) = tso(iia,1)
          hi(ia,1) = ho(iia,1)
          rhogi(ia,1) = psi(ia,1) * 144.0 / (766.55232 * tsi(ia,1))
          rholi(ia,1) = 0.0
          rhoi(ia,1) = rhogi(ia,1)
          vi(ia,1) = 144.0 * mdot(il) / (a(ia,1) * rhoi(ia,1))
          asvi = (dummy * dabs(tsi(ia,1)))**0.5
          machi(ia,1) = vi(ia,1)/asvi
        if (debug .eq. 8.0 .or. debug .ge. 10.0) then
          write(6,9990) ia,psi(ia,1),tsi(ia,1),hi(ia,1)
          write(6,9991) vi(ia,1),asvi,machi(ia,1)
          write(6,9992) rhoi(ia,1),rholi(ia,1),rhogi(ia,1)
9990  format('node # =',3x,i3,3x,'ps, ts, h (in) =',3(2x,f10.4))
9991  format('vel, a, m (in) =',3(2x,f10.4))
9992  format('rho(in) =',3(2x,f10.4))
      endif
*
*   CALCULATE THE HEAT TRANSFER COEFFICIENT FOR 100% GAS NODES
*   BASED ON INLET PRESSURE
*
call value(n13,p13,stemp,psi(ia,1),ts)

```

```

tdim = (tsi(ia,1)-ts)/(tcnst-ts)
  if (tdim .lt. 0.0) tdim = 0.0
call value2(n14,num1,p14,t14,cp,gpsi(ia,1),tdim,cp)
call value2(n14,num1,p14,t14,thkg,psi(ia,1),tdim,cond)
call value2(n14,num1,p14,t14,viscg,psi(ia,1),tdim,visgfi)
ti = tsi(ia,1) * (1.0 + 0.2 * machi(ia,1) ** 2.0)
pi(ia,1) = psi(ia,1) * (1.0 + 0.2 * machi(ia,1) ** 2.0)
  pdi(ia,1) = pi(ia,1) - psi(ia,1)
hgl = hgfi - hlf1
*
dtemp = 0.0
dpress = 0.0
re = 48.0 * mdot(il) /(pyi * visgfi * exdi)
call htcoeff(phase,dtemp,dpress,dp,cond,visgfi,rholfi,
1           rhogfi,st,hgl,re,xf,exdi,qdott)
if (debug .eq. 8.0 .or. debug .ge. 10.0) then
  write(6,9993) cp,cond,visgfi
  write(6,9994) ti,pi(ia,1),pdi(ia,1),re,qdott,
1           twall(ia),tave(ia,1)
9993  format('cp, k, mu =',3(2x,f10.4))
9994  format('To, Po, Pd(in) =',3(2x,f12.4),/,
1           'Re, Hc ,Twall, Tave =',4(2x,f12.4))
endif
*
* CALCULATE THE OUTLET PROPERTIES OF THE GAS NODE
*
do nn1 = 1, 2
  if(tave(ia,1) .le. 0.0) tave(ia,1) = ti
  if(nn1 .eq. 1) tave(ia,1) = ti
  dtemp = twall(ia) - tave(ia,1)
*
  to = ti + (pyi * length(ia) * qdott * (twall(ia)-ti) *
1           exdi / (mdot(il) * cp * 144.0))
  if (to .lt. 0.0) to = ti/2.0
  call htandf(machi(ia,1),ti,to,twall(ia),macho(ia,1))
  ratio = (1+0.2*machi(ia,1)**2.0)/(1+0.2*macho(ia,1)**2.0)
  tso(ia,1) = tsi(ia,1) * (to/ti) * ratio
  pso(ia,1) = psi(ia,1) * (machi(ia,1)/macho(ia,1)) *
1           (tso(ia,1)/tsi(ia,1))**0.5
  vo(ia,1) = vi(ia,1) * (machi(ia,1)/macho(ia,1)) *
1           (tso(ia,1)/tsi(ia,1))**0.5
  po(ia,1) = pi(ia,1) * (tso(ia,1)/tsi(ia,1)) * ratio**3.5
  pdo(ia,1) = po(ia,1) - pso(ia,1)
  ho(ia,1) = hi(ia,1) + cp * (to-ti)
  rhoo(ia,1) = rhoi(ia,1) * pso(ia,1) * tsi(ia,1) /

```



```

3          pi(12,1)
116      format ('iteration # ',i3,2x,'time =',f8.3,3x,'mass =',
1           f8.3,4x,'mdot =',f10.5,',pst =',f8.3,4x,'qerr =',
2           f8.3,4x,'pes =',f8.3,/,6(3x,f8.3),/,6(3x,f8.3))
2         ENDIF
if (debug .eq. 9.0) then
  write(6,117) l,qdt,qdt1,qerr
  do jj = 1,m
    write(6,118) jj,twall(jj),q(jj,1),q(jj,2),delta(jj),
1           tsi(jj,1),tso(jj,1),tsi(jj,2),tso(jj,2),
2           qdtt(jj,1),qdtt(jj,2)
  enddo
117      format ('iteration # ',i3,2x,'Gas Qdot =',f8.3,3x,
1           'Liq Qdot =',f8.4,3x,'Q error =',f8.4,/,t3,'Node #',
2           t13,'T wall',t23,'Gas Q',t33,'Liq Q',t43,'Delta Q',t53,
3           'G Tsi',t63,'G Tso',t73,'L Tsi',t83,'L Tso',t93,'G Hc',
4           t103,'L Hc',/)
118      format (6x,i4,10(2x,f8.4))
  endif
*****  

*   CONDITION TO END THE QDOT CONVERGENCE LOOP *
*  

* IF (dabs(qerr) .le. qdterr .or. flag1 .ge. m8) goto 200  

*  

* End of L do loop (The Calculated Qdot to the Nodal      *  

* Model is Within the Error Margin of PERRMX      *  

* to the past Qdot)      *  

*  

* ENDDO  

*****  

200  do ii = 1,m1
  nn = m - ii
  qdot(nn,1) = qdot(nn+1,1) + q(nn,1)
  qdot(nn,2) = qdot(nn+1,2) - q(nn,2)
  enddo
c
c  DETERMINE THE EXIT PLANE CONDITIONS
c
j2 = 0
if (va .eq. 1) ax = sa
PXS=PSO(1,1)
205  IF(PXS.LT.PTP) PXS=PTP
      IF(PXS.GT.PST) PXS=PST
      call value(n8,p8,lentp,PXS,hlx)
      call value(n9,p9,gentp,PXS,hgx)

```

```

call value(n10,p10,lent,PXS,slx)
call value(n11,p11,gent,PXS,sgx)
call value(n4,p4,lrhop,PXS,rholx)
call value(n5,p5,grhop,PXS,rhogx)
YX=(SO(1,1)-SLX)/(SGX-SLX)
HX=YX*HGX+(1.0-YX)*HLX
VX=SQRT(2.0*32.2*777.649*ABS(HTOTE(1)-HX))
RHOX=144.0*MDOT(1)/(VX*AX)
RHOXV=YX/(1.0/RHOX-1.0/RHOLX*(1.0-YX))
call value(n6,rhovp,p6,RHOXV,pxsc)
    if (j2 .eq. 0) psx1 = pxsc
    gamma = dabs(pxs - psx1)
    psx1 = pxs
    gerr = pxsc - pxs
    IF (dabs(gerr) .gt. .0.001) then
        if (gerr .gt. 0.0) then
            pxs = pxs + gamma/2.0
        else
            pxs = pxs - gamma/2.0
        endif
        j2 = j2 + 1
        if (j2.gt.150) go to 210
        go to 205
    endif
*
210  IF (debug .eq. 7.0 .or. debug .ge. 10.0) THEN
    write (6,115) j2,pxs,time,gerr,mass,slx,sgx,rhox,yx,sx,hx
115  format ('counter = ',i4,4x,'PXS = ',f8.3,3(3x,f8.3),/,
1           6(3x,f8.3))
    ENDIF
*
if (prop.eq.1.0) then
    call H2SAT(pxs,tx)
else
    call o2sat(pxs,tx)
endif
PXD=144.0/(2.0*RHOX*32.2)*(MDOT(1)/AX)*(MDOT(1)/AX)
PX=PXS+PXD
FM=MDOT(1)*VX/32.2
FP=PXS*AX
THRUST=FM+FP
ISP=THRUST/MDOT(1)
    dpx = pxs - ptp
    flag = 0.0
    flag1 = 0.0

```

```

    flag3 = 0.0
    IF (dpx .le. delptp) then
        exit = 1
    ENDIF
*
*      Print Output
*
do l1 = 1, 2
if (l1 .eq. 1) then
    write (6,1000) subhd,subt
    write (6,1430)
    write (6,1100) name(1),mdot(l1),name(2),isp,name(3),thrust,
1           name(4),qerr,name(5),rhot,name(6),pst,name(7),
2           pes(l1),name(8),htot,name(9),a(1,l1)
    write (6,1200)
    write (6,1300) pxs,tx,rhox,yx,hx,vx
    write (6,1400) pes(l1),tes(l1),rhoe(l1),ye(l1),htote(l1)
else
    write (6,1000) subhd,subt
    write (6,1460)
    write (6,1100) name(1),mdot(l1),name(2),isp,name(3),thrust,
1           name(4),qerr,name(5),rhot,name(6),pst,name(7),
2           pes(l1),name(8),htot,name(9),a(1,l1)
    write (6,1200)
    write (6,1400) pes(l1),tes(l1),rhoe(l1),ye(l1),htote(l1)
endif
do nc1 = 1,m
    write (6,1500) nc1,ps0(nc1,l1),tsi(nc1,l1),rho0(nc1,l1),
1           yo(nc1,l1),ho(nc1,l1),vo(nc1,l1),qdot(nc1,l1),
2           a(nc1,l1),phisq(nc1,l1),length(nc1),dpf(nc1,l1),
3           dpm(nc1,l1),dpt(nc1,l1),xtt(nc1,l1),rn(nc1,l1),
4           qdtt(nc1,l1)
enddo
    write (6,1600) pst,tliq,rholt,yt,hs(l1)
enddo
*
    write (6,1000) subhd,subt
    write (6,1700) ps,tsp,tts,mdot(1)
do l1 = 1, 2
if (l1 .eq. 1) then
    write (6,1430)
    write (6,1800)
else
    write (6,1460)
    write (6,1800)

```

```

endif
do nc2 = 1,m
write (6,1900) nc2,pi(nc2,11),psi(nc2,11),pdi(nc2,11),
1      po(nc2,11),ps0(nc2,11),pdo(nc2,11),tsi(nc2,11),
2      tso(nc2,11),twall(nc2),tave(nc2,11),q(nc2,11),
3      delta(nc2),k(nc2,11)
enddo
enddo
*
1000 format('1',//,18a4,/,10a8,/)
1100 format(9(4x,a8,f9.4,/,))
1200 format(t2,'NODE #',t11,'STATIC P',t23,'TEMP',t32,'DENSITY',
1      t42,'QUALITY',t51,'ENTHALPY',t61,'VELOCITY',t73,'QDOT',
2      t83,'AREA',t93,'PHISQ',t102,'LENGTH',t114,'DPF',t124,'DPM',
3      t134,'DPT',t144,'Xtt',t153,'Re #',t160,'Ht Trans C')
1300 format(/,t2,'OUTLET',t11,f9.4,4(x,f9.4),x,f9.2)
1400 format(/,t2,'EXIT',t11,f9.4,3(x,f9.4),x,f9.2)
1430 format(/,t2,'FLOW PROPERTIES OF THE VENT PART OF THE SYSTEM')
1460 format(/,t2,'FLOW PROPERTIES OF THE HEAT EXCHANGER PART OF',
1      'THE SYSTEM')
1500 format(t4,i3,t11,f9.4,7(x,f9.4),x,f9.2,5(x,f9.4),x,f9.2,x,f9.4)
1600 format(/,t2,'INLET',t11,f9.4,5(x,f9.4))
1700 format(/,t2,'Ptank static',t21,f9.4,/,t2,'Pchoke',t21,f9.4,/,t2,
1      'Tchoke',t21,f9.4,/,t2,'Mdot',t21,f9.4)
1800 format(/,t2,'NODE #',t13,'Pt in',t25,'Ps in',t34,'Pd in',
1      t44,'Pt out',t53,'Ps out',t63,'Pd out',t73,'Ts in',t83,
2      'Ts out',t93,'T wall',t103,'T ave',t113,'Qdot',t123,
3      'Del Q',t133,'K loss',/)
1900 format(t4,i3,t11,f9.4,12(x,f9.4))

STOP
END

subroutine H2SAT(PST,tsat)
c
double precision pst,tsat,fp,t
c
FP=LOG(PST/187.506)
T=1.00003+FP*(2.12094E-1+FP*(2.83129E-2+FP*1.75686E-3))
TSAT=59.3568*T
return
END

subroutine h2SVEL(Y,PS,RSL,RSV,as)
c
double precision ps,rsl,rsv,as,p,fpf,tst1,ts,tsm,tc,tfun,t,tk

```

```

double precision pa,dpvdt,d2pvt1,d2pvdt,csatt1,csatt2,csat
double precision drodt1,tt,drodt2,drodt,rho,dsdrho,dmdp1,dmdp

c
P=PS/187.506264
FPF=LOG(P)
TST1=FPF*(2.83128E-2+FPF*(1.75686E-3))
TS=1.00002+FPF*(2.12094E-1+TST1)
TSM=TS*32.976
TC=32.976
TFUN=(TC-TSM)**0.33333
T=TSM*1.8
IF(Y.LE.0.0) Y=0.0
TK=T/1.8
PA=10.0**2.00062-50.0970/(TK+1.0044)+1.74849E-2*TK)*14.696
DPVDT=PA*2.30258*(50.0970/(TK+1.0044)**2+1.74849E-2)/1.8
D2PVT1=DPVDT**2/PA
D2PVDT=D2PVT1+PA*2.30258/1.8**2*(-2.0*50.0970/(TK+1.0044)**3)
CSATT1=1.68157*TK/(32.976-TK)**0.1-32.8027
CSATT2=TK*(3.35743E-2+TK*(-7.68297E-4+6.90292E-6*TK))
CSAT=(CSATT1+TK*(6.81698+TK*(-0.731943+CSATT2)))/2.01572
DRODT1=-0.38* 7.32346E-3/(32.976-TK)**0.62+4.40742E-4
TT=(32.976-TK)**0.33333
DRODT2=TT*TT*(1.66667*2.92263E-04-2.0*4.00849E-05*TT)
DROLDT=(DRODT1-1.33333*6.62079E-4*TT+DRODT2)*69.9099
RHO=1.0/(1.0/RSL+Y*(1.0/RSV-1.0/RSL))
DSDRHO=-1.0/RHO/RHO*DPVDT*144.0
DSDP1=777.649/144.0*CSAT/(T*DPVDT)+1.0/RSL/RSL*DROLDT
DSDP=DSDP1+(1.0/RHO-1.0/RSL)*D2PVDT/DPVDT
AS=SQRT(ABS(-DSDRHO/DSDP*32.2))
return
END

subroutine htcoeff(phase,dt,dp,cp,k,mu,rhol,rhov,st,lambda,re,x,
1          diam,qdot)
c
c This subroutine calculates the two phase heat transfer coefficient
c
double precision cp,k,mu,rhol,rhov,st,x
double precision lgx(12),lgf(12),lgsf(12),sf(12),f,s,c2,d3
real lambda
c
data lgx/-1.0,-0.824,-0.745,-0.699,-0.585,0.0,0.176,0.398,0.663,
1      0.778,1.778,2.0/
data lgf/0.0,0.0212,0.0414,0.0569,0.1139,0.4472,0.5563,0.699,
1      0.8751,0.9542,1.699,1.8692/

```

```

data lgsf/0.176,0.415,0.672,1.0,1.19,1.29,1.398,1.544,1.602,
1      1.663,1.778,2.0/
data sf/0.84,0.73,0.58,0.37,0.25,0.2,0.16,0.12,0.11,0.1,0.09,
1      0.09/
c
dtemp = abs(dt)
dpress = abs(dp)
gc = 32.2
a1 = .023
a2 = re**0.8
a3 = (cp * mu * 3600. / k)**0.4
a4 = (k * 12.) / (3600. * diam)
hfc = a1 * a2 * a3 * a4
if (phase .lt. 1.5 .or. phase .ge. 3.0) then
    qdot = hfc
c   write(6,*) cp,mu,k,diam,a1,a2,a3,a4,qdot
      return
endif
c   write(6,*) a1,a2,a3,a4
c
b1 = .00122
b2 = ((k/3600.)**0.79) * (cp**0.45) * (rholf**0.49) * (gc**0.25) *
1      (dtemp**0.24) * (dpress**0.75)
b3 = ((st*12.)**0.5) * (mu**0.29) * (lambda**0.24) * (rhov**0.24)
hfz = b1 * b2 / b3
c   write(6,*) b1,b2,b3
c
c1 = 1 / x
c2 = log10(c1)
if (c2 .gt. 2.0) c2 = 2.0
call value(12,lgx,lgf,c2,f)
f = 10.0**f
c   write(6,*) c1,c2,f
c
d1 = re * f**1.25
d2 = d1 / 10000.0
d3 = log10(d2)
if (d3 .gt. 2.0) d3 = 2.0
call value(12,lgsf,sf,d3,s)
c   write(6,*) d1,d2,d3
    qdot = hfc*f + hfz*s
c   write (6,100) b2,b3,c1,c2,d1,d2
c   write (6,100) re,hfc,f,hfz,s,qdot
c100 format (3(3x,f14.4),/3(3x,f14.4))
      return

```

END

subroutine INTL(PO,K,MDOT,A,RHO,pin)
double precision po,rho,pin,dpf

c
real k,mdot

c
DPF=144.0*K*((MDOT/A)**2.0)/(2.0*RHO*32.2)
PIN=PO+DPF
return
END

subroutine spray(mdot,psmi,dsm,ed,pback)

c
c subroutine spray.f to model flow in the spray injection tube
c

character*1 label

double precision rod(12),lod(12),roverd,loverd,pback
real mdin(200),mdout(200),mds(200),as(200),vel(200)
real pin(200),pout(200),pnode(200),ptank(200),x(200),delp(200)
real node(10),asec(10)

real subhd(18),xtitl(18),yp(18),ymdot(18),ymds(18),

1 yvel(18),yas(18)

real mdot,mdoti,mdoto,mdots,mdotp

real k fsm,k bsm,k csm,k bsi,ks

integer*2 ibox,iloc

common /contrl/ibox,iloc

data nb/12/rod/1.,1.5,2.,3.,4.,6.,8.,10.,12.,14.,16.,20./

data lod/20.,14.,12.,12.,14.,17.,24.,30.,34.,38.,42.,50./

ibox = 1

iloc = 0

*

open (unit=8, file='spray.data',status='unknown')

*

read (8,100) label

read (8,*) pu,zliq,ztank,dmdot

read (8,100)

read (8,*) zsm,dsi,zsi,n,nbar

read (8,100)

read (8,*) ks,cds,acc,rho,visc,roverd

read (8,100)

read (8,*) tol,nlim

read (8,100)

read (8,*) opn,opt,omdin,omds,ovel,oas

read (8,101) subhd,xtitl,yp,ymdot,ymds,yvel,yas

```

read (8,*) nsec
do i = 1,nsec
    read (8,*) node(i),asec(i)
enddo
100 format (//a1)
101 format (18a4/18a4/18a4/18a4/18a4/18a4)
    write (6,1)
1 format (5x,'SPRAY MANIFOLD AND INJECTION TUBE FLOW MODEL')
write (6,2) zliq,pu,psmi,acc,rho,zsm,dsm
2 format (5x,'Liquid Level          = ',f6.1,' in')
1   5x,'Ullage Pressure        = ',f6.2,' psia'
2   5x,'Spray Manifold Inlet Pressure = ',f6.3,' psia'
3   5x,'Acceleration Level      = ',f6.1,' g'
4   5x,'Liquid Density          = ',f6.3,' lbm/ft3'
5   5x,'Spray Manifold Tube Length = ',f6.1,' in'
6   5x,'Spray Manifold Tube ID    = ',f6.2,' in')
dzsi = zsi/n
write (6,3) nbar,zsi,dsi,n,dzsi,ks
3 format (5x,'Number of Spray Injection Tubes = ',i6/
1   5x,'Spray Injection Tube Length  = ',f6.1,' in'
2   5x,'Spray Injection Tube ID     = ',f6.3,' in'
3   5x,'Number of Orifices         = ',i6/
4   5x,'Orifice Spacing            = ',f6.2,' in'
5   5x,'Orifice Loss Coefficient   = ',f6.2/)
pu = 144.*pu
psmi = 144.*psmi
tol = 144*tol
zliq = zliq/12.
ztank = ztank/12.
dsm = dsm/12.
dsi = dsi/12.
zsm = zsm/12.
zsi = zsi/12.
dzsi = dzsi/12.
asm = 3.14159*dsm*dsm/4.
asi = 3.14159*dsi*dsi/4.
do j = 1,nsec
    n1 = n2 + 1
    n2 = node(j)
    do i = n1,n2
        as(i) = asec(j)/144.
    enddo
enddo
kcsrm = .5*(1. - (dsi/dsm)**2)
call value(nb,rod,rod,roverd,loverd)

```

```

zu = ztank - zliq
gc = 32.2
do 10 j = 1,nlim
    qsm = (mdot/asm)**2/(2.*rho*gc)
    call frict(dsm,ed,mdot,visc,fsm)
    k fsm = fsm*zsm/dsm
    k bsm = fsm*loverd
    dp fsm = qsm*(k fsm + k bsm + k csm)
    dp f1 = qsm*(k fsm)
    ph sm = rho*acc*zsm
    psmo = psmi - dp fsm - ph sm
    pback = psmi - dp f1
    pback = pback/144.0
    dp sm = dp fsm + ph sm
    mdot i = mdot/nbar
    q i = (mdot i /asi)**2/(2.*rho*gc)
    call frict(dsi,ed,mdot i,visc,fsi)
    k bsi = fsi*loverd
    dp si = q i*k bsi
    pi = psmo - dp si
    do i = 1,n
        x(i) = i*dzsi - dzsi/2.
        aorf = as(i)
        z = ztank - zsi + x(i)
        if (z .lt. zu) pt = pu
        if (z .ge. zu) pt = pu + rho*acc*(z - zu)
        call pres(i,n,pi,po,pn,pt,dpf,ph,mdot i,mdot o,mdots,
1           dzsi,dsi,ed,asi,aorf,ks,rho,visc,acc)
        if (aorf .le. 0.0) vel(i) = 0.0
        if (aorf .gt. 0.0) vel(i) = mdots/(rho*cds*aorf)
        pin(i) = pi
        pout(i) = po
        pnode(i) = pn
        ptank(i) = pt
        mdin(i) = mdot i
        mdout(i) = mdot o
        mds(i) = mdots
        pi = pout(i)
        mdot i = mdout(i)
        if (i .lt. n) dpsi = dpsi + dpf - ph
        if (i .eq. n) dpsi = dpsi + dpf - ph/2.
    enddo
    ptcal = pn - ks/(2.*rho*gc)*(mdots/aorf)**2
    delpt(j) = ptcal - pt
    if (j .eq. 1) then

```

```

if (delpt(j) .lt. 0.) mdot = mdot - dmdot
if (delpt(j) .gt. 0.) mdot = mdot + dmdot
else
  prod = delpt(j)*delpt(j-1)
  if (delpt(j) .lt. 0. .and. prod .gt. 0.) mdot = mdot - dmdot
  if (delpt(j) .gt. 0. .and. prod .gt. 0.) mdot = mdot + dmdot
  if (delpt(j) .lt. 0. .and. prod .lt. 0.) then
    dmdot = dmdot/2.
    mdot = mdot - dmdot
  endif
  if (delpt(j) .gt. 0. .and. prod .lt. 0.) then
    dmdot = dmdot/2.
    mdot = mdot + dmdot
  endif
endif
10 continue
if (abs(delpt(j)) .gt. tol) write (6,1002) delpt(j)
1002 format ('*** tank pressure does not converge, delpt = ',
1      f8.4,' psi ***')
15 dppump = (psmi - pt)/144.
dpsm = dpsm/144.
dpsi = dpsi/144.
write (6,200) mdot,dppump,dpsm,dpsi
200 format (5x,'Pump Flow Rate           = ',f6.3,' lbm/sec'
1      5x,'Pump Pressure Rise       = ',f6.3,' psi'
2      5x,'Spray Manifold Tube Delta p = ',f6.3,' psi'
3      5x,'Spray Injection Tube Delta p = ',f6.3,' psi//')
write (6,4)
4   format (5x,'Node',2x,'Distance',3x,'Inlet',2x,'Outlet',
1      3x,'Nodal',3x,'Tank',5x,'Inlet',4x,'Outlet',
2      3x,'Injection',3x,'Injection',3x,'Orifice')
write (6,5)
5   format (24x,'p',7x,'p',7x,'p',7x,'p',
1      7x,'mdot',5x,'mdot',7x,'mdot',6x,'Velocity',
2      5x,'CdA')
write (6,6)
6   format (13x,'(in)',4x,'(psia)',2x,'(psia)',2x,'(psia)',2x,'(psia)',
1      2x,'(lbm/sec)',1x,'(lbm/sec)',1x,'(lbm/sec)',5x,'(fps)',
2      6x,'(in2)')

do i = 1,n
  pin(i) = pin(i)/144.
  pout(i) = pout(i)/144.
  pnode(i) = pnode(i)/144.
  ptank(i) = ptank(i)/144.
  as(i) = 144.*as(i)

```

```

        x(i) = 12.*x(i)
        write (6,210) i,x(i),pin(i),pout(i),pnode(i),ptank(i),
1           mdin(i),mdout(i),mds(i),vel(i),as(i)
210    format (5x,i3,4x,f6.2,3x,4(f6.3,2x),
1           3(e9.3,1x),4x,f5.2,4x,f7.5)
       enddo
       if (opn .eq. 0.) go to 20
       call crtplt(-11,11,2,00,0,1,subhd,xtitl,yp,0.,0.,
1           0.,0.,n,x,pnode,0,0)
20    if (opt .eq. 0.) go to 21
       call crtplt(0,0,0,01,0,0,0,0,0,0,0.,
1           0.,0.,n,x,ptank,0,0)
21    if (omdin .eq. 0.) go to 22
       call crtplt(-11,11,2,00,0,1,subhd,xtitl,ymdot,0.,0.,
1           0.,0.,n,x,mdin,0,0)
22    if (omds .eq. 0.) go to 23
       call crtplt(-11,11,2,00,0,1,subhd,xtitl,ymds,0.,0.,
1           0.,0.,n,x,mds,0,0)
23    if (ovel .eq. 0.) go to 24
       call crtplt(-11,11,2,00,0,1,subhd,xtitl,yvel,0.,0.,
1           0.,0.,n,x,vel,0,0)
24    if (oas .eq. 0.) go to 25
       call crtplt(-11,11,2,00,0,1,subhd,xtitl,yas,0.,0.,
1           0.,0.,n,x,as,0,0)
25    return
       end

subroutine TPHS(HI,RHOL8,RHOG8,MDOT,AN8,HL8,HG8,SL8,SG8,
1           y8,h8,s8,rho8,p8d)
double precision hi,rhol8,rhog8,hl8,hg8,sl8,sg8,h8,rho8,p8d
real k,mdot
c
c
K=0.41405*(MDOT*MDOT)/(AN8*AN8)
A1=K*(1.0/RHOG8-1.0/RHOL8)*(1.0/RHOG8-1.0/RHOL8)
B1=HG8-HL8+2.0*K/RHOL8*(1.0/RHOG8-1.0/RHOL8)
C1=HL8-HI+K/(RHOL8*RHOL8)
Y8=(-B1+SQRT(ABS(B1*B1-4.0*A1*C1)))/(2.0*A1)
IF(Y8.LE.0.00) then
    y8=0.00
endif
IF(Y8.GE.1.0) then
    y8=1.00
endif
H8=Y8*HG8+(1.0-Y8)*HL8

```

```

S8=Y8*SG8+(1.0-Y8)*SL8
RHO8=1.0/(1.0/RHOL8+Y8*(1.0/RHOG8-1.0/RHOL8))
P8D=144.0/(2.0*32.2*RHO8)*(MDOT/AN8)*(MDOT/AN8)
return
END

```

```
    subroutine TPS(A,G,HI,HO,K,P,YO,RLO,RGO,YI,RLI,RGI,MDT,PHISQ,b,
```

```
1           d,e,f)
```

```
double precision p,rlo,rgo,rli,rgi,phisq,b,d,e,f
double precision dpf1,dpf,vo,vi,dpm,dpt,ph,pinlet
real k,mdt
```

```
c
```

```
RHOO=1.0/(1.0/RLO+YO*(1.0/RGO-1.0/RLO))
```

```
RHOI=1.0/(1.0/RLI+YI*(1.0/RGI-1.0/RLI))
```

```
RHO=(RHOO+RHOI)/2.0
```

```
YA=(YI+YO)/2.0
```

```
IF(YA.GT.1.0) then
```

```
    ya=0.999
```

```
endif
```

```
RHOLA=(RLI+RLO)/2.0
```

```
aa = ((1.0-ya)**2.0)*PHISQ/(RHOLA)
```

```
bb = 1.0 / rho
```

```
cc = aa
```

```
if (bb .ge. aa) cc = bb
```

```
DPF1=(MDT/A)**2
```

```
DPF=144.0*K*(DPF1)*cc/(2.0*32.2)
```

```
B=DPF
```

```
VO=MDT/(RHOO*A)*144.0
```

```
VI=MDT/(RHOI*A)*144.0
```

```
DPM=ABS(MDT/A*(VO-VI)/32.2)
```

```
DPT=DPM+DPF
```

```
PH=G*RHO*(HI-HO)/1728.0
```

```
PINLET=P+DPT-PH
```

```
D=PINLET
```

```
E=DPT
```

```
F=DPM
```

```
return
```

```
END
```

```
subroutine value(np,x,y,xin,yout)
```

```
c
```

```
c This subroutine performs lagrangian interpolation within a set
```

```
c of (x,y) pairs to give yout corresponding to xin
```

```
c
```

```
double precision x(np),y(np),xin,yout
```

```

c
if (xin .le. x(1)) yout = y(1)
if (xin .le. x(1)) return
if (xin .ge. x(np)) yout = y(np)
if (xin .ge. x(np)) return
do 10 i = 1,np
    k = i
    if (xin .lt. x(i)) go to 30
10 continue
30 ffr = (xin - x(k - 1))/(x(k) - x(k - 1))
    yout = y(k - 1) + ffr*(y(k) - y(k - 1))
c   write (6,*) np,x,y,xin,yout
return
end

SUBROUTINE VALUE2(NPX,NPY,X,Y,Z,XIN,YIN,ZOUT)
c   DIMENSION X(NPX),Y(NPY),Z(NPX,NPY)
      double precision X(NPX),Y(NPY),Z(NPX,NPY),XIN,YIN,ZOUT
C
IF(XIN .LE. X(1) .AND. YIN .LE. Y(1))ZOUT=Z(1,1)
IF(XIN .LE. X(1) .AND. YIN .LE. Y(1))RETURN
C
IF(XIN .GE. X(NPX) .AND. YIN .LE. Y(1))ZOUT=Z(NPX,1)
IF(XIN .GE. X(NPX) .AND. YIN .LE. Y(1))RETURN
C
IF(XIN .LE. X(1) .AND. YIN .GE. Y(NPY))ZOUT=Z(1,NPY)
IF(XIN .LE. X(1) .AND. YIN .GE. Y(NPY))RETURN
C
IF(XIN .GE. X(NPX) .AND. YIN .GE. Y(NPY))ZOUT=Z(NPX,NPY)
IF(XIN .GE. X(NPX) .AND. YIN .GE. Y(NPY))RETURN
C
IF(XIN .GT. X(1))GO TO 30
DO 20 I=1,NPY
    M=I
    IF(YIN .LT. Y(I))GO TO 25
20 CONTINUE
25 FFRY=(YIN-Y(M-1))/(Y(M)-Y(M-1))
    ZOUT=Z(1,M-1)+FFRY*(Z(1,M)-Z(1,M-1))
    RETURN
C
30 IF(XIN .LT. X(NPX))GO TO 60
DO 50 I=1,NPY
    M=I
    IF(YIN .LT. Y(I))GO TO 55
50 CONTINUE

```

```

55 FFRY=(YIN-Y(M-1))/(Y(M)-Y(M-1))
      ZOUT=Z(NPX,M-1)+FFRY*(Z(NPX,M)-Z(NPX,M-1))
      RETURN
C
60 IF(YIN .GT. Y(1))GO TO 90
      DO 80 I=1,NPX
          L=I
          IF(XIN .LT. X(I))GO TO 85
80 CONTINUE
85 FFRX=(XIN-X(L-1))/(X(L)-X(L-1))
      ZOUT=Z(L-1,1)+FFRX*(Z(L,1)-Z(L-1,1))
      RETURN
C
90 IF(YIN .LT. Y(NPY))GO TO 120
      DO 110 I=1,NPX
          L=I
          IF(XIN .LT. X(I))GO TO 115
110 CONTINUE
115 FFRX=(XIN-X(L-1))/(X(L)-X(L-1))
      ZOUT=Z(L-1,NPY)+FFRX*(Z(L,NPY)-Z(L-1,NPY))
      RETURN
C
120 DO 130 I=1,NPX
      L=I
      IF(XIN .LT. X(I))GO TO 135
130 CONTINUE
135 FXR=(XIN-X(L-1))/(X(L)-X(L-1))
      DO 140 I=1,NPY
          M=I
          IF(YIN .LT. Y(I))GO TO 145
140 CONTINUE
145 FYR=(YIN-Y(M-1))/(Y(M)-Y(M-1))
C
      ZXLO=Z(L-1,M-1)+FYR*(Z(L-1,M)-Z(L-1,M-1))
      ZXHI=Z(L,M-1)+FYR*(Z(L,M)-Z(L,M-1))
C
      ZOUT=ZXLO+FXR*(ZXHI-ZXLO)
      RETURN
      END

SUBROUTINE VALUE3(NPX,MAXY,NPY,X,Y,Z,XIN,YIN,ZOUT)
integer npy(maxy)
double precision X(NPX),Y(NPX,MAXY),Z(NPX,maxy),xin,yin,zout
c double precision X(15),Y(15,15),Z(15,15),xin,yin,zout
dimension a(50),b(50),c(50),d(50)

```

```

C
IF(XIN .LE. X(1) .AND. YIN .LE. Y(1,1)) then
    ZOUT=Z(1,1)
    RETURN
endif

C
IF(XIN .GE. X(NPX) .AND. YIN .LE. Y(npx,1)) then
    ZOUT=Z(NPX,1)
    RETURN
endif

C
IF(XIN .LE. X(1) .AND. YIN .GE. Y(1,NPY(1))) then
    ZOUT=Z(1,NPY(1))
    RETURN
endif

C
IF(XIN .GE. X(NPX) .AND. YIN .GE. Y(npx,NPY(npx))) then
    ZOUT=Z(NPX,NPY(npx))
    RETURN
endif

C
IF(XIN .GT. X(1))GO TO 30
DO I=1,NPY(1)
    M=I
    IF(YIN .LT. Y(1,I))GO TO 25
    enddo
25 FFRY=(YIN-Y(1,M-1))/(Y(1,M)-Y(1,M-1))
    ZOUT=Z(1,M-1)+FFRY*(Z(1,M)-Z(1,M-1))
    RETURN

C
30 IF(XIN .LT. X(NPX))GO TO 60
DO I=1,NPY(npx)
    M=I
    IF(YIN .LT. Y(npx,I))GO TO 55
    enddo
55 FFRY=(YIN-Y(npx,M-1))/(Y(npx,M)-Y(npx,M-1))
    ZOUT=Z(NPX,M-1)+FFRY*(Z(NPX,M)-Z(NPX,M-1))
    RETURN

C
60 DO I=1,NPX
    L=I
    IF(XIN .LT. X(I))GO TO 85
    enddo
85 do j = 1, npy(l)
    a(j) = y(l,j)

```

```

        b(j) = z(l,j)
enddo
call value(npv(l),a,b,yin,z1)
do k = 1, npv(l-1)
    c(k) = y(l-1,k)
    d(k) = z(l-1,k)
enddo
call value(npv(l-1),c,d,yin,z2)
FFRX=(XIN-X(L-1))/(X(L)-X(L-1))
ZOUT=Z2+FFRX*(Z1-Z2)
RETURN
END

subroutine XPARAM(YO,VLO,VGO,RLO,RGO,YI,VLI,VGI,RLI,RGI,x)
c
double precision vlo,vgo,rlo,rgo,vli,vgi,rgi,rgi,x
double precision visla,visga,rhola,rhoga,x1
c
YA=(YI+YO)/2.0
VISLA=(VLI+VLO)/2.0
VISGA=(VGI+VGO)/2.0
RHOGA=(RGI+RGO)/2.0
RHOLA=(RLI+RLO)/2.0
IF(YA.LE.1.0E-06) then
    ya=1.0e-06
endif
IF(YA.GE.1.0) then
    ya=1.0
endif
X1=((ABS(VISLA/VISGA))**0.1)*(((1.0-YA)/YA)**0.9)
X=X1*SQRT(ABS(RHOGA/RHOLA))
c   write (6,10) yi,yo,ya,vli,vlo,visla,vgi,vgo,visga,rgi,rgo,rhoga,
c   1           rli,rlo,rhola,x1,x
c   10 format (6(3x,f8.4),/6(3x,f8.4),/3(3x,f8.4),2(3x,f10.4))
      return
END

```

3.4.2 Integrated Zero-g TVS Model

```

c
c program tvs.f to model zero-g Thermodynamic Venting System (TVS)
c transient performance
c
character*1 label
real ptitl(18,4),subhd(18),xtitl(18),ketitl(4,2),
1    ypu(18),ytu(18),ymu(18),ypl(18),ytl(18),yml(18),

```

```

2 ytw(18),ymwl(18),ymdv(18),ymdsu(18),ymddu(18),ymdbw(18),
3 ymdif(18),ymds(18),ynpump(18),ydppmp(18)
real tsat(25),psat(25),enthf(25),enthg(25),
1 shpf(25),densf(25),condf(25),viscf(25),texpf(25)
real pvap(8),tvap(8),tnrm(11),enth(8,11),shv(8,11),
1 shp(8,11),dens(8,11),cond(8,11),visc(8,11)
real tal(12),shpal(12),node(10),dorf(10)
real as(100),ddrop(100),ad(100),vd(100),veld(100),ndrop(100)
real ton(100),toff(100),tcyc(100),pton(100),
1 mvent(100),mvnt(100),sfc(100)
real mup,mlp,mwlp,mvp,mdvp
real mdsp,mdslp,mdsup,mddup,mdbwp,mdlup,npumpp
real mu,ml,mwl,mw,mv,mui,mli,mwli,mvi,mwlmax
real kwu,muwu,kuwl,muuwl,kwl,muwl,muul,kul
real kus,muus,cls,muls,kud,muud,nud
real ks,mds,mdsl,mdsu,mddu,mdsw,mdbw,mdbwmx,mdlu,mdul
real mdv,mdvent,mdp,mdpump,mdsne,mdsul,mdcond,mdcndp,mdsi
real mdotd,npumpd,npump,npumpi,nmax,nmin,ip
integer jsymb(2)
integer*2 ibox,iloc
common /pltcom/misc(3),nc,miss(13),dclim,ltick,nfig,nptmin,
1 nlines,nchlin,ptitl
common /contrl/ibox,iloc
common /sprayin/dsm,zsm,dsi,zsi,norf,nbar,
1 ks,cds,as,vd,roverd,dmdot,tol,nlim,
2 rhos,viscs
common /pumpin/npumpd,nmin,nmax,hpid,hpmotr,effp,ip,dm
common /tankdim/dtank,hcyl,hbulk,tkw,c1
common /tankvol/v1,v2,vbi,vbo,vci,vco
common /tankarea/ab,ac,at
common /tankout/j,nline,iprint,timep(3000),
1 pup(3000),tup(3000),vup(3000),mup(3000),twp(3000),
2 plp(3000),tlp(3000),vlp(3000),mlp(3000),
3 pwlp(3000),twlp(3000),vwlp(3000),mwlp(3000),
4 mvp(3000),tsp(3000),mdlup(3000),mdsp(3000),mdslp(3000),
5 mdsup(3000),mddup(3000),mdbwp(3000),mdulp(3000),mdcndp(3000),
6 qwup(3000),quwlp(3000),qwlp(3000),qulp(3000),qudp(3000),
7 qusp(3000),qlsp(3000),npumpp(3000),dpprmp(3000)
nc = 2
nfig = 0
nlines = 4
ibox = 257
jsymb(1) = 40
jsymb(2) = 43
data ketitl/4hEVAP,4hORAT,4hTION,4h ,
```

```

1 4hCOND,4hENSA,4hTION,4h  /
data ru,rhow,gc,ed/766.0,176.3,32.2,1.0e-6/
data tal/0.0,25.0,50.0,75.0,100.0,125.0,150.0,180.0,260.0,
1 485.0,760.0,1000.0/nal/12/
data shpal/0.0,0.001,0.0075,0.02,0.0525,0.09,0.11,0.125,0.15,
1 0.2,0.225,0.25/
c
c read input data for tank model
c
read (5,100) label
read (5,*) xd,xchar,he,xcond,prtsp,outp
read (5,100) label
read (5,*) pui,tui,pli,twi,twli,full,xl1
read (5,100) label
read (5,*) dtank,hcyl,hbulk,tkw,dsb,d1,d2
read (5,100) label
read (5,*) mdvent,mdsi,dthex,qflux,g,hwliq
read (5,100) label
read (5,*) pmin,pmax,delt2,iprnt2,iplot2
read (5,100) label
read (5,*) fintim,delt1,xdelt1,iprnt1,iplot1,nline
read (5,100) label
read (5,*) opu,otu,omu,opl,otl,oml
read (5,100) label
read (5,*) otw,omwl,omdv,omdsu,omddu,omdbw
read (5,100) label
read (5,*) omdlu,omdul,omds,onpump,odppmp
read (5,200) subhd,xtitl,ypu,ytu,ymu,ypl
read (5,200) ytl,yml,ytw,ymwl,ymdv,ymdsu
read (5,200) ymddu,ymdbw,ymdif,ymds,ynpump,ydppmp
read (5,201) ptitl

c
c read input data for pump model
c
read (5,100) label
read (5,*) mdotd,dpd,npumpi,npumpd,xhp,xn
read (5,100) label
read (5,*) deltat,effp

c
c read input data for spray manifold/injection tubes model
c
read (5,100) label
read (5,*) dsm,zsm,dsi,zsi,norf,nbar
read (5,100) label
read (5,*) ks,cds,roverd,dmdot,tol,nlim

```

```

read (5,*) nsec
do i = 1,nsec
    read (5,*) node(i),dorf(i)
enddo
100 format (//a1)
200 format (18a4/18a4/18a4/18a4/18a4/18a4)
201 format (18a4)
c
c read LH2 saturation properties
c
open(unit=2,file='h2prop',status='old')
read (2,*) nsat
do i = 1,nsat
    read (2,*) tsat(i),psat(i),enthf(i),dumvar,shpf(i),densf(i),
1           texpf(i),condf(i),viscf(i)
    read (2,*) enthg(i)
enddo
c
c read GH2 properties as a function of pressure and temperature
c
read (2,*) np,nt
read (2,*) (tnrm(i),i=1,nt),tcnst
do i = 1,np
    read (2,*) pvap(i),tvap(i)
    do j = 1,nt
        read (2,*) enth(i,j),shv(i,j),shp(i,j),dens(i,j),
1           cond(i,j),visc(i,j)
    enddo
enddo
if (outp .eq. 1.0) then
write (6,1)
1 format (5x,'TANK DIMENSIONS')
write (6,2) dtank,tkw,hcyl,hbulk
2 format (5x,'Tank Diameter          ',f6.1,' in')
1     5x,'Tank Wall Thickness      ',f6.2,' in'
2     5x,'Cylinder Height         ',f6.1,' in'
3     5x,'Bulkhead Height         ',f6.1,' in//)
write (6,3)
3 format (5x,'SPRAY MANIFOLD/INJECTION TUBE DIMENSIONS')
write (6,4) zsm,dsm,zsi,dsi,nbar,norf,ks,cds
4 format (5x,'Spray Manifold Tube Length   ',f6.1,' in')
1     5x,'Spray Manifold Tube ID     ',f6.3,' in'
2     5x,'Spray Injection Tube Length ',f6.1,' in'
3     5x,'Spray Injection Tube ID     ',f6.3,' in'
4     5x,'Number of Spray Injection Tubes ',i6/

```

```

5      5x,'Number of Orifices          ',i6/
6      5x,'Orifice Loss Coefficient   ',f6.2/
7      5x,'Orifice Discharge Coefficient ',f6.2/
write (6,5)
5 format (t2,'Time',t10,'pU',t18,'TU',t26,'VU',t34,'MU',
1      t42,'pL',t50,'TL',t58,'VL',t66,'ML',t74,'TW',
2      t82,'TWL',t90,'Npump',t98,'dppump',t106,'dTpump',
3      t114,'HPO',t122,'mdS',t130,'mdSU',t138,'mdDU',
4      t146,'mdBW',t154,'mdLU',t162,'mdcond')
write (6,6)
6 format (t2,'sec',t9,'psia',t19,'R',t25,'ft3',t33,'lbf',
1      t41,'psia',t51,'R',t57,'ft3',t65,'lbf',t75,'R',
2      t83,'R',t91,'rpm',t99,'psid',t109,'R',
3      t114,'HP',t120,'lbf/sec',t128,'lbf/sec',t136,'lbf/sec',
4      t144,'lbf/sec',t152,'lbf/sec',t160,'lbf/sec')
endif
do j = 1,nsec
    node1 = node2 + 1
    node2 = node(j)
    do i = node1,node2
        ds = dorf(j)/12.0
        as(i) = 3.14159*ds**2/4.0
        ddrop(i) = xd*ds
        ad(i) = 3.14159*ddrop(i)**2
        vd(i) = 3.14159*ddrop(i)**3/6.0
    enddo
enddo
dtank = dtank/12.0
hcyl = hcyl/12.0
hbulk = hbulk/12.0
htank = hcyl + 2.0*hbulk
tkw = tkw/12.0
dsb = dsb/12.0
dsm = dsm/12.0
zsm = zsm/12.0
dsi = dsi/12.0
zsi = zsi/12.0
xl1 = xl1/12.0
d1 = d1/12.0
d2 = d2/12.0
dzsi = zsi/norf
dchar = xchar*dtank
c
c initial ullage and liquid masses
c

```

```

call volarea(vt)
dtanki = dtank - 2.0*tkw
call value(nsat,psat,tsat,pli,tli)
call value(nsat,tsat,densf,tli,rhol)
call value(nsat,tsat,densf,twli,rhowl)
vli = full/100.0*vt
mli = rhol*vli
vui = vt - vli
mui = 144.0*pui*vui/(ru*tui)
ts = tli
tsw = ts
c
c in 1 g, set maximum thickness of wall liquid layer to 0.01 in
c due to liquid run-off and calculate maximum wall liquid mass
c
call area(vli,y,awu,aul,awl,hliq,hu)
if (g .ge. 1.0) mwlmmax = rhol*awu*0.01/12.0
c
c pump design conditions
c
dm = 720.0/(3.14159*npumpd)*(2.0*144.0*dpd/rhol)**0.5
hpid = xhp*mdotd*dpd*144.0/(550.0*effp*rhol)
hpmotr = hpid
nmax = npumpd*(1.0 + xn)
nmin = npumpd*(1.0 - xn)
ip = 6.018e+05*hpid*deltat/(npumpd*(nmax - nmin))/2.0
if (pui .ge. pmax) mdp = mdsi
if (pui .le. pmin) mdp = 0.0
mdpump = mdsi
c
c time integration of variables
c
i = 0
j = 0
85 if (npump .gt. 0.0) then
    delt = delt1
    if (pu .le. pl) delt = xdel1*delt1
    iprint = iprnt1/xdel1
    iplot = iplot1
else
    delt = delt2
    iprint = iprnt2
    iplot = iplot2
endif
mu = mui + dmudt*delt

```

```

ml = mli + dmldt*delt
if (ml .le. 0.0) ml = 0.0
mwli = mwli + dmwldt*delt
if (mwli .le. 0.0) mwli = 0.0
if (g .ge. 1.0 .and. mwli .ge. mwli max) mwli = mwli max
mv = mvi + mdv*delt
vl = vli + dvldt*delt
if (vl .le. 0.0) vl = 0.0
vwli = mwli/rhowl
vu = vt - vl - vwli
tl = tli + dtldt*delt
tu = tui + dtudt*delt
call value(nsat,psat,tsat,pu,tusat)
if (tu .le. tusat) tu = tusat
twli = twli + dtwldt*delt
if (twli .ge. tusat) twli = tusat
if (twli .le. ts) twli = ts
tw = twi + dtwdt*delt
qpump = qpumpi + hpo*0.707*delt
mui = mu
mli = ml
mwli = mwli
mvi = mv
vli = vl
tui = tu
tli = tl
twli = twli
twi = tw
qpumpi = qpump
c
c ullage, bulk liquid, and wall liquid pressures
c
pu = mu*ru*tu/(144.0*vu)
call value(nsat,tsat,psat,tl,pl)
call value(nsat,tsat,psat,twl,pwl)
c
c pump control logic
c
if (pu .ge. pmax) flag1 = 1.0
if (pu .le. pmin) flag1 = 0.0
c
c performance calculations
c
if (flag1 .eq. 0.0) then
  if (cyc .eq. 0.0) ncyc = ncyc + 1

```

```

toff(ncyc) = time - timeon
timeof = time
cyc = 1.0
endif
if (flag1 .eq. 1.0) then
    ton(ncyc) = time - timeof
    tcyc(ncyc) = ton(ncyc) + toff(ncyc)
    pton(ncyc) = 100.0*ton(ncyc)/tcyc(ncyc)
    sfc(ncyc) = 3600.0*mv/time
    mvent(ncyc) = mv
    pup(ncyc) = pu
    tup(ncyc) = tu
    mup(ncyc) = mu
    plp(ncyc) = pl
    tlp(ncyc) = tl
    mlp(ncyc) = ml
    twp(ncyc) = tw
    mdsp(ncyc) = mds
    timeon = time
    cyc = 0.0
endif
c
c pump model
c
call value(nsat,tsat,shpf,tl,cpl)
nmdot = 0
87 call pump(flag1,npump,npumpi,delt,mdpump,rhol,cpl,dppump,dtpump,
1 dnt,hpo)
if (dppump .le. 0.01) mdp = 0.0
if (dppump .gt. 0.01) mdp = mdpump
tpump = tl + dtpump
c
c vent control logic
c
psatp = -36.37 + 4.6054*tpump - 0.20369*tpump*tpump
1 + 0.0031745*tpump*tpump*tpump
if (psatp .gt. pmin) flag2 = 1.0
if (psatp .le. pmin) flag2 = 0.0
mdv = 0.0
if (flag1 .gt. 0.0 .and. flag2 .gt. 0.0) mdv = mdvent
c
c pressure drop in the recirculation line (between the pump outlet
c and spray manifold inlet)
c
if (mdp .le. 0.0) dprec = 0.0

```

```

call htc(tu,tl,dtanki,cpul,rhoul,betaul,kul,muul,g,hul)
qul = hul*aul*(tu - tl)

c
c ullage-to-droplet
c
td = ts
tud = (tu + td)/2.0
tudn = (tud - tusat)/(tcnst - tusat)
call value2(np,nt,pvap,turm,dens,pu,tudn,rhoud)
call value2(np,nt,pvap,turm,visc,pu,tudn,muud)
call value2(np,nt,pvap,turm,cond,pu,tudn,kud)
qud = 0.0
if (mdp .le. 0.0) go to 17
do io = 1,norfu
    red = rhoud*veld(io)*ddrop(io)/muud

c
c textbook heat-transfer correlation for liquid droplets (Kreith)
c
nud = 0.3125*red**0.602
if (ddrop(io) .gt. 0.0) hud = nud*kud/ddrop(io)
qud = qud + ndrop(io)*hud*ad(io)*(tu - td)
enddo
17 call value(nsat,tsat,densf,td,rhod)

c
c ullage-to-spray bar
c
tus = (tu + ts)/2.0
betaus = 1.0/tus
tusn = (tus - tusat)/(tcnst - tusat)
call value2(np,nt,pvap,turm,shp,pu,tusn,cpus)
call value2(np,nt,pvap,turm,dens,pu,tusn,rhous)
call value2(np,nt,pvap,turm,cond,pu,tusn,kus)
call value2(np,nt,pvap,turm,visc,pu,tusn,muus)
call htc(tu,ts,dsb,cpus,rhous,betaus,kus,muus,g,hus)
aus = 3.14159*dsb*hu
qus = nbar*hus*aus*(tu - ts)
qus = 0.0

c
c liquid-to-spray bar
c
tls = (tl + ts)/2.0
call value(nsat,tsat,shpf,tls,cpls)
call value(nsat,tsat,densf,tls,rhol)
call value(nsat,tsat,texpf,tls,betals)
call value(nsat,tsat,condf,tls,kls)

```

```

call value(nsat,tsat,viscf,ts,muls)
call htc(tl,ts,dsb,cpls,rhol,beals,kls,muls,g,hls)
als = 3.14159*dsb*hliq
qls = nbar*hls*als*(tl - ts)
qls = 0.0
c
c environment-to-wall
c
qew = qflux*awu
qel = qflux*awl
c
c mass-transfer rates
c
c droplet-to-ullage boil-off
c
call value(nsat,psat,enthf,pu,hf)
call value(nsat,psat,enthg,pu,hgsat)
hfgu = hgsat - hf
mddu = (qud/3600.0 - mdsu*cpl*(tusat - ts))/hfgu
if (mddu .lt. 0.0) mddu = 0.0
if (mddu .lt. 0.0) qud = 3600.0*mdsu*cpl*(tusat - ts)
if (mddu .lt. 0.0) ts = tusat - qud/(3600.0*mdsu*cpl)
if (mddu .gt. mdsu) mddu = mdsu
if (mddu .gt. mdsu) qud = mdsu*(hfgu + cpl*(tusat - ts))*3600.0
c
c non-evaporated spray droplet
c
mdsne = mdsu - mddu
if (g .ge. 1.0) then
    mdsw = 0.0
    mdsul = mdsne
else
    mdsw = mdsne
    mdsul = 0.0
endif
c
c droplet boil-off from wall
c
call value(nsat,tsat,enthf,tl,hf)
call value(nsat,tsat,enthg,tl,hg)
hfgl = hg - hf
dpudt = pu*(dmudt/mu + dtudt/tu - dvudt/vu)
if (mw1 .le. 0.0) mdbw = 0.0
if (mw1 .gt. 0.0) then
    if (pu .gt. pw1) mdbw = 0.0

```

```

if (pu .lt. pw1 .and. dpudt .gt. 0.0)
1   mdbw = ((qwl + quwl)/3600.0 - mdsu*cpl*(tw1 - tsw))/hfgl
    if (pu .lt. pw1 .and. dpudt .le. 0.) then
        dtdp = 0.37781 - 4.9170e-3*pwl + 21.7623e-6*pwl*pwl
        mdbw = ((qwl + quwl)/3600.0 - mdsu*cpl*(tw1 - tsw))
1   - mw1*cpl*dtdp*dpudt)/hfgl
    endif
endif
mdbwmx = mw1/delt
if (mdbw .ge. mdbwmx) mdbw = mdbwmx
c
c liquid-to-ullage boil-off
c
if (pu .gt. pl) mdlu = 0.0
if (pu .le. pl .and. dpudt .gt. 0.0)
1   mdlu = (ql/3600.0 - mds1*cpl*(tl - ts))/hfgl
    if (pu .le. pl .and. dpudt .le. 0.0) then
        dtdp = 0.37781 - 4.9170e-3*pl + 21.7623e-6*pl*pl
        mdlu = (ql/3600.0 - mds1*cpl*(tl - ts))
1   - ml*cpl*dtdp*dpudt)/hfgl
    endif
c
c condensation
c
if (tu .gt. tusat) mdul = 0.0
if (tu .le. tusat) mdul = (qud + qu1 + quwl)/(3600.0*hfgu)
if (he .eq. 0.0 .and. flag1 .eq. 1.0)
1   mdcond = xcond*qul/(3600.0*hfgu)
if (flag1 .eq. 0.0) mdcond = 0.0
if (he .eq. 1.0) mdcond = 0.0
c
c rates of change of ullage, wall liquid, and bulk liquid masses
c
dmudt = mddu + mdbw + mdlu - mdul - mdcond
dmwldt = mdsu - mdbw
dmldt = mds1 + mdul + mdcond + mdsul - mdlu - mds - mdv
if (g .ge. 1.0 .and. mw1 .ge. mw1max) dmldt = dmldt + dmwldt
c
c rates of change of ullage, liquid, and wall liquid volumes
c
call value(nsat,tsat,densf,tl,rhol)
dvldt = dmldt/rhol
if (mw1 .le. 0.0) dvwldt = 0.0
if (mw1 .gt. 0.0) dvwldt = dmwldt/rhowl
dvudt = -dvldt - dvwldt

```

```

c
c rates of change of temperature
c
c ullage
c
qu = qwu - quwl - qul - qud - quis
if (mdcond .eq. 0.0) qu = qwu - quwl - qud - quis
enthu = dmudt*hgsat
tun = (tu - tusat)/(tcnst - tusat)
call value2(np,nt,pvap,tncst,shv,pu,tun,cvu)
dtudt = (qu/3600.0 - 144.0/778.0*pu*dvudt + enthu
1      - cvu*tu*dmudt)/(mu*cvu)

c
c bulk liquid
c
ql = qel + qul - qls
if (mddu .lt. 0.0) td = ts
if (mddu .ge. 0.0) td = tusat
if (ml .gt. 0.0) dtldt = (ql/3600.0 - mdu*hfgl
1      + mdsul*cpl*(td - tl) - mdsu*cpl*(tl - ts))/(ml*cpl)
if (ml .le. 0.0) dtldt = 0.0

c
c wall liquid
c
qudchk = 3600.0*mdsu*cpl*(tusat - ts)
if (qud .gt. qudchk) tsw = tusat
if (qud .le. qudchk) tsw = ts + qud/3600.0/((mdsu + 0.0001)*cpl)
if (mwl .gt. 0.0) dtwldt = ((qwl + quwl)/3600.0
1      - mdsw*cpl*(twl - tsw))/(mwl*cpl)
if (mwl .le. 0.0) dtwldt = 0.0

c
c tank wall
c
call vwall(hliq,vw)
mw = rhow*vw
call value(nal,tal,shpal,tw,cpw)
qw = qew - qwu - qwl
dtwdt = qw/3600.0/(mw*cpw)

c
c output listing
c
if (outp .eq. 0.0) go to 19
if (mod(i,iprint) .eq. 0.0)
1 write (6,7) time,pu,tu,vu,mu,pl,tl,vl,ml,tw,ts,npump,
2      dppump,dtpump,hpo,mds,mdsu,mddu,mdbw,mdlu,mdcond

```

```

7   format (f7.0,f6.2,9(2x,f6.2),2x,f6.1,
1      3(2x,f6.3),6(1x,f7.4))
if (mod(i,iplot) .ne. 0.0) go to 19
j = j + 1
timep(j) = time
pup(j) = pu
tup(j) = tu
vup(j) = vu
mup(j) = mu
twp(j) = tw
plp(j) = pl
tlp(j) = tl
vlp(j) = vl
mlp(j) = ml
pwlp(j) = pwl
twlp(j) = twl
vwlp(j) = vwl
mwlp(j) = mw1
mvp(j) = mv
mdvp(j) = mdv
tsp(j) = ts
mdlup(j) = mdu
mdsp(j) = mds
mdslp(j) = mdsl
mdsup(j) = mdsu
mddup(j) = mddu
mdbwp(j) = mdbw
mdulp(j) = mdul
mdcndp(j) = mdcond
qwup(j) = qwu
quwlp(j) = quwl
qwlp(j) = qwl
qulp(j) = qul
qudp(j) = qud
qusp(j) = quis
qlsp(j) = qls
npumpp(j) = npump
dppmp(j) = dppump
19 i = i + 1
time = time + delt
if (time .le. fintim) go to 85
if (ton(ncyc) .eq. 0.0) ncyc = ncyc - 1
write (6,999)
999 format ('1')
do i = 1,ncyc

```

```

ston = ston + ton(i)
stoff = stoff + toff(i)
stcyc = stcyc + tcyc(i)
if (i .eq. 1) mvnt(i) = mvent(i)
if (i .gt. 1) mvnt(i) = mvent(i) - mvent(i-1)
write (6,1000) i,ton(i),toff(i),tcyc(i),pton(i),mvnt(i),sfc(i)
1000 format (5x,'Cycle No.          = ',i12/
  1      5x,'On Time           = ',f12.3,' sec'
  2      5x,'Off Time          = ',f12.3,' sec'
  3      5x,'Cycle Time         = ',f12.3,' sec'
  4      5x,'% On Time          = ',f12.3,' %'
  5      5x,'Vented Mass        = ',f12.3,' lbm'
  6      5x,'Specific Fuel Consumption = ',f12.3,' lbm/hr')
write (6,1001) pup(i),tup(i),mup(i),plp(i),tlp(i),mlp(i),
  1      twp(i),mdsp(i)
1001 format (5x,'Ullage pressure     = ',f12.3,' psia'
  1      5x,'Ullage temperature   = ',f12.3,' R'
  2      5x,'Ullage mass         = ',f12.3,' lbm'
  3      5x,'Liquid pressure      = ',f12.3,' psia'
  4      5x,'Liquid temperature    = ',f12.3,' R'
  5      5x,'Liquid mass         = ',f12.3,' lbm'
  6      5x,'Wall temperature      = ',f12.3,' R'
  7      5x,'Pump flow rate       = ',f12.3,' lbm/sec')
enddo
ptonav = 100.0*ston/stcyc
sfcav = 3600.0*mvent(ncyc)/stcyc
write (6,1002) ston,stoff,stcyc,ptonav,mvent(ncyc),sfcav
1002 format (5x,'On Time           = ',f12.3,' sec'
  1      5x,'Off Time          = ',f12.3,' sec'
  2      5x,'Cycle Time         = ',f12.3,' sec'
  3      5x,'% On Time          = ',f12.3,' %'
  4      5x,'Vented Mass        = ',f12.3,' lbm'
  5      5x,'Specific Fuel Consumption = ',f12.3,' lbm/hr')
c
c output listing
c
call prtout
c
c output plotting
c
if (opu .eq. 1.0) call crtplt(-11,12,2,00,0,0,subhd,xtitl,
  1      ypu,0.,0.,0.,0.,j,timep,pup,0,0)
if (opl .eq. 1.0) call crtplt(12,12,2,00,0,0,subhd,xtitl,
  1      ypl,0.,0.,0.,0.,j,timep,plp,0,0)
if (otu .eq. 1.0) call crtplt(21,00,2,00,0,0,subhd,xtitl,

```

```

1      ytu,0.,0.,0.,j,timep,tup,0,0)
if (otl .eq. 1.0) call crtplt(22,00,2,00,0,0,subhd,xtitl,
1      ytl,0.,0.,0.,j,timep,tlp,0,0)
if (omu .eq. 1.0) call crtplt(-11,22,2,00,0,0,subhd,xtitl,
1      ymu,0.,0.,0.,j,timep,mup,0,0)
if (oml .eq. 1.0) call crtplt(12,00,2,00,0,0,subhd,xtitl,
1      yml,0.,0.,0.,j,timep,mlp,0,0)
if (omwl .eq. 1.0) call crtplt(21,00,2,00,0,0,subhd,xtitl,
1      ymw1,0.,0.,0.,j,timep,mwlp,0,0)
if (omdsu .eq. 1.0) call crtplt(-11,22,2,00,0,0,subhd,xtitl,
1      ymdsu,0.,0.,0.,j,timep,mdsup,0,0)
if (omddu .eq. 1.0) call crtplt(12,00,2,00,0,0,subhd,xtitl,
1      ymddu,0.,0.,0.,j,timep,mddup,0,0)
if (omdbw .eq. 1.0) call crtplt(21,00,2,00,0,0,subhd,xtitl,
1      ymdbw,0.,0.,0.,j,timep,mdbwp,0,0)
if (omdlu .eq. 1.0) call crtplt(22,00,2,10000,jsymb(1),0,subhd,
1      xtitl,ymdif,0.,0.,0.,j,timep,mdlup,0,0)
if (omdul .eq. 1.0) call crtplt(00,00,0,10001,jsymb(2),0,0,0,
1      0.,0.,0.,0.,j,timep,mdulp,0,0)
if (omdlu .eq. 1.0 .and. omdul .eq. 1.0)
1  call crtkey(2,jsymb,ketitl,-1,-1)
if (omds .eq. 1.0) call crtplt(-11,12,2,00,0,0,subhd,xtitl,
1      ymds,0.,0.,0.,j,timep,mdsp,0,0)
if (omdv .eq. 1.0) call crtplt(12,12,2,00,0,0,subhd,xtitl,
1      ymdv,0.,0.,0.,j,timep,mdvp,0,0)
if (onpump .eq. 1.0) call crtplt(-11,12,2,00,0,0,subhd,xtitl,
1      ynpump,0.,0.,0.,j,timep,npumpp,0,0)
if (otw .eq. 1.0) call crtplt(21,00,2,00,0,0,subhd,xtitl,
1      ytw,0.,0.,0.,j,timep,twp,0,0)
if (odppmp .eq. 1.0) call crtplt(12,12,2,00,0,0,subhd,xtitl,
1      ydppmp,0.,0.,0.,j,timep,dppmp,0,0)
stop
end

```

subroutine area(v1,y,awu,aul,awl,z,hu)

c
c subroutine area.f to calculate the heat transfer areas,
c and liquid and gas heights of an elliptical bulkhead tank
c

```

real 1
common /tankdim/do,l,ho,t,c1
common /tankvol/v1,v2,vbi,vbo,vci,vco
common /tankarea/ab,ac,at
data tol,nlim/.01,40/
r = do/2. - t

```

```

h = ho - t
vb = vbi
vc = vci
c
c liquid level is in upper bulkhead
c
if (vl .gt. v2) then
  v = vl - vb - vc
  do i = 1,nlim
    fy = y*y*y - 3.*h*h*y + 3.*v*h*h/(3.14159*r*r)
    dfy = 3.*y*y - 3.*h*h
    dely = fy/dfy
    y = y - dely
    if (abs(dely) .le. tol) go to 5
  enddo
5  z = y + h + 1
  awl = ab + ac
1  + 3.14159*(y*(c1*y*y + r*r)**.5 + r*r/c1**.5
2  *(log(y*c1**.5 + (c1*y*y + r*r)**.5) - log(r)))
  rul = r*(1. - y/h)
  aul = 3.14159*rul*rul
c
c liquid level is in cylindrical segment
c
else
if (vl .gt. v1) then
  z = (vl - vb)/(3.14159*r*r) + h
  awl = ab + 2.*3.14159*r*(z - h)
  rul = r
  aul = 3.14159*rul*rul
c
c liquid level is in lower bulkhead
c
else
if (vl .gt. 0.) then
  v = vb - vl
  do i = 1,nlim
    fy = y*y*y - 3.*h*h*y + 3.*v*h*h/(3.14159*r*r)
    dfy = 3.*y*y - 3.*h*h
    dely = fy/dfy
    y = y - dely
    if (abs(dely) .le. tol) go to 10
  enddo
10   z = h - y
  else

```

```

z = 0.
endif
awl = 3.14159*(h*(c1*h*h + r*r)**.5
1   - (h - z)*(c1*(h - z)*(h - z) + r*r)**.5
2   + r*r/c1**.5*(log(h*c1**.5 + (c1*h*h + r*r)**.5)
3   - log((h - z)*c1**.5 + (c1*(h - z)*(h - z) + r*r)**.5)))
rul = z*r/h
aul = 3.14159*rul*rul
endif
endif
awu = at - awl
dul = 2.*rul
hu = 2.*h + 1 - z
return
end

subroutine frict(d,ed,mdot,visc,f)
c
c subroutine frict.f to calculate the friction coefficient
c for flow in a pipe
c
real mdot
re = 4.*mdot/(3.14159*visc*d)
if (re .lt. 2300.) f = 64./(re + 1.)
if (re .gt. 2300.)
1 f = 0.25/(log10(ed/3.7 + 2.51/(re*sqrt(.0056 +
2 .5/(re**.32)))))**2
return
end

subroutine htc(t1,t2,d,cp,rho,beta,k,mu,a,h)
c
c This subroutine computes the free convection heat-transfer
c coefficient for horizontal and vertical surfaces
c
real k,mu,nu
if (d .eq. 0.) h = 0.
if (d .eq. 0.) return
ra = 3600*32.2*a*beta*abs(t1 - t2)*d**3*rho*rho*cp
1 /(mu*k)
nu = 0.555*ra**0.25 + 0.447
h = nu*k/d
return
end

```

```

subroutine pdrop(mdot,rho,mu,l1,d1,d2,dptot)
c
c subroutine pdrop.f to calculate the pressure drops
c between the pump outlet and spray manifold inlet
c
real reno(9),kloss1(9),kloss2(9)
real mdot,mu,l1,kb1,kb2,kb3,kb4,kc,kflm
data reno/1.5e+5,2.0e+5,3.0e+5,4.0e+5,6.0e+5,8.0e+5,1.0e+6,
1 2.0e+6,3.0e+6/
data kloss1/0.32,0.26,0.21,0.19,0.173,0.168,0.163,0.160,0.158/
data kloss2/0.20,0.15,0.14,0.128,0.12,0.118,0.117,0.115,0.114/
ed = 1.0e-6
gc = 32.2
a1 = 3.14159*d1*d1/4.0
a2 = 3.14159*d2*d2/4.0
c
c 90-degree bend at pump outlet
c
re1 = 4.0*mdot/(3.14159*mu*d1)
call value(9,reno,kloss1,re1,kb1)
c
c straight section downstream of 90-degree bend
c
call frict(d1,ed,mdot,mu,f1)
k1 = f1*l1/d1
c
c reducer
c
kc = 0.5*(1.0 - (d2/d1)**2)
c
c 132.5-degree bend
c
re2 = 4.0*mdot/(3.14159*mu*d2)
call value(9,reno,kloss1,re2,kb2)
kb2 = 1.22*kb2
c
c flowmeter
c
kflm = 1.308
c
c 95.5-degree bend downstream of flowmeter
c
call value(9,reno,kloss2,re2,kb3)
kb3 = 1.03*kb3
c

```

```

c 48-degree bend
c
c   call value(9,reno,kloss2,re2,kb4)
    kb4 = 0.66*kb4
c
c   pressure drops
c
const = mdot*mdot/(2.0*rho*gc*144.0)
dpb1 = kb1*const/a1**2
dp1 = k1*const/a1**2
dpc = kc*const/a2**2
dpb2 = kb2*const/a2**2
dpflm = kflm*const/a2**2
dpb3 = kb3*const/a2**2
dpb4 = kb4*const/a2**2
dptot = dpb1 + dp1 + dpc + dpb2 + dpflm + dpb3 + dpb4
c   write (6,12) dpb1,dp1,dpc,dpb2,dpflm,dpb3,dpb4,dptot
c12  format (3x,'dpb1 = ',f8.5,3x,'dp1 = ',f8.5,3x,'dpc = ',f8.5/
c   1      3x,'dpb2 = ',f8.5,3x,'dpflm = ',f8.5,3x,'dpb3 = ',f8.5/
c   2      3x,'dpb4 = ',f8.5,3x,'dptot = ',f8.5)
      return
      end

subroutine prtout
c
c subroutine prtout.f to print the complete output of
c program tvs.f in a file named outdat
c
real mu,ml,mwl,mv
real mdlu,mds,mdsl,mdsu,mddu,mdbw,mdul,mdcond,npump
common /tankout/nstep,nline,iprint,time(3000),
1      pu(3000),tu(3000),vu(3000),mu(3000),tw(3000),
2      pl(3000),tl(3000),vl(3000),ml(3000),
3      pwl(3000),twl(3000),vwl(3000),mwl(3000),
4      mv(3000),ts(3000),mdl(3000),mds(3000),mdsl(3000),
5      mdsu(3000),mddu(3000),mdbw(3000),mdul(3000),mdcond(3000),
6      qwu(3000),quwl(3000),qwl(3000),ql(3000),qud(3000),
7      qu(3000),qls(3000),npump(3000),dppump(3000)
open (unit=15,file='outdat')
icount = nstep/(nline*iprint) + 1
do j = 1,icount
    imin = imax + 1
    imax = j*nline*iprint
    if (imax .gt. nstep) imax = nstep
    write (15,1)

```

```

1   format ('1',t6,'Time',t18,'pU',t28,'TU',t38,'VU',t48,'MU',
1       t58,'TW',t68,'pL',t78,'TL',t88,'VL',t98,'ML',
2       t107,'pWL',t117,'TWL',t127,'VWL',t137,'MWL',
3       t148,'MV',t158,'TS',t166,'mdLU')
    write (15,2)
2   format (t7,'sec',t16,'psia',t29,'R',t37,'ft3',t47,'lbm',
1       t59,'R',t66,'psia',t79,'R',t87,'ft3',t97,'lbm',
2       t106,'psia',t119,'R',t127,'ft3',t137,'lbm',
3       t147,'lbm',t159,'R',t163,'lbm/sec')
    do 20 i = imin,imax
        if (mod(i-1,iprint) .ne. 0.0) go to 20
        write (15,3) time(i),pu(i),tu(i),vu(i),mu(i),
1           tw(i),pl(i),tl(i),vl(i),ml(i),
2           pwl(i),twl(i),vwl(i),mwl(i),
3           mv(i),ts(i),mdlu(i)
3   format (f9.1,15(1x,f9.3),1x,e9.3)
c3   format (f10.1,15(1x,e10.4))
20 continue
    write (15,4)
4   format ('1',t6,'Time',t17,'mdS',t26,'mdSL',t36,'mdSU',
1       t46,'mdDU',t56,'mdBW',t66,'mdUL',t74,'mdCOND',
2       t87,'qWU',t96,'qUWL',t107,'qWL',t117,'qUL',
3       t127,'qUD',t137,'qUS',t147,'qLS',
4       t155,'Npump',t164,'dppump')
    write (15,5)
5   format (t7,'sec',t13,'lbm/sec',t23,'lbm/sec',t33,'lbm/sec',
1       t43,'lbm/sec',t53,'lbm/sec',t63,'lbm/sec',t73,'lbm/sec',
2       t84,'Btu/hr',t94,'Btu/hr',t104,'Btu/hr',t114,'Btu/hr',
3       t124,'Btu/hr',t134,'Btu/hr',t144,'Btu/hr',
4       t157,'rpm',t167,'psi')
    do 30 i = imin,imax
        if (mod(i-1,iprint) .ne. 0.0) go to 30
        write (15,6) time(i),mds(i),mdsl(i),mdsu(i),
1           mddu(i),mdbw(i),mdul(i),mdcond(i),
2           qwu(i),quwl(i),qwl(i),qul(i),
3           qud(i),qus(i),qls(i),
4           npump(i),dppump(i)
6   format (f9.1,7(1x,e9.3),9(1x,f9.3))
30 continue
enddo
return
end

subroutine spray(pman,mdoti,pull,zu,zliq,ztank,dchar,rhol,acc,ed,
1               time,prtsp,mdot,mdsu,mdsl,veld,ndrop,norfu)

```

```

c
c subroutine spray.f to model flow in the spray injection tube
c
real rod(12),lod(12),roverd,loverd
real mdin(100),mdout(100),mds(100),as(100),asp(100)
real vd(100),veld(100),ndrop(100)
real pin(100),pout(100),pnode(100),ptank(100),x(100),delpt(200)
real mdot,mdoti,mdotsi,mdoto,mdots,mdsu,mdsl
real k fsm,k bsm,k csm,k bsi,ks
common /sprayin/dsm,zsm,dsi,zsi,n,nbar,
1           ks,cds,as,vd,roverd,dmdt,tol,nlim,
2           rho,visc
data nb/12/rod/1.,1.5,2.,3.,4.,6.,8.,10.,12.,14.,16.,20./
data lod/20.,14.,12.,12.,14.,17.,24.,30.,34.,38.,42.,50./
psmi = 144.*pman
pu = 144.*pull
dmdot = dmdt
dzsi = zsi/n
asm = 3.14159*dsm*dsm/4.
asi = 3.14159*dsi*dsi/4.
k csm = .5*(1. - (dsi/dsm)**2)
call value(nb,rod,lad,roverd,loverd)
gc = 32.2
mdot = mdoti
do 10 j = 1,nlim
    qsm = (mdot/asm)**2/(2.*rho*gc)
    call frict(dsm,ed,mdot,visc,fsm)
    k fsm = fsm*zsm/dsm
    k bsm = fsm*loverd
    dp fsm = qsm*(k fsm + k bsm + k csm)
    phsm = rho*acc*zsm
    dpsm = dp fsm + phsm
    psmo = psmi - dpsm
    mdotsi = mdot/nbar
    qi = (mdotsi/asi)**2/(2.*rho*gc)
    call frict(dsi,ed,mdotsi,visc,fsi)
    k bsi = fsi*loverd
    pi = psmo - qi*k bsi
    mdsu = 0.
    mdsl = 0.
    norfu = 0
    do i = 1,n
        x(i) = i*dzsi - dzsi/2.
        aorf = as(i)
        z = ztank - zsi + x(i)

```

```

if (z .lt. zu) pt = pu
if (z .ge. zu) pt = pu + rhol*acc*(z - zu)
call pres(i,n,pi,po,pn,pt,dpf,ph,mdotsi,mdoto,mdots,
1      dzsi,dsi,ed,asi,aorf,ks,rho,visc,acc)
pin(i) = pi
pout(i) = po
pnode(i) = pn
ptank(i) = pt
mdin(i) = mdotsi
mdout(i) = mdoto
pi = pout(i)
mdotsi = mdout(i)
mds(i) = mdots
if (aorf .gt. 0.0) veld(i) = mdots/(rho*cds*aorf)
if (aorf .le. 0.0) veld(i) = 0.0
if (veld(i) .gt. 0.0)
1      ndrop(i) = nbar*mdots*dchar/(rho*vd(i)*veld(i))
if (veld(i) .le. 0.0) ndrop(i) = 0.0
if (z .lt. zu) norfu = norfu + 1
if (z .lt. zu) mdsu = mdsu + mdots
if (z .ge. zu) mdsl = mdsl + mdots
enddo
mdsu = nbar*mdsu
mdsl = nbar*mdsl
ptcal = pn - ks/(2.*rho*gc)*(mdots/aorf)**2
delpt(j) = ptcal - pt
dpt = delpt(j)/144.0
if (abs(dpt) .lt. tol) go to 15
if (j .eq. 1) then
  if (delpt(j) .lt. 0.) mdot = mdot - dmdot
  if (delpt(j) .gt. 0.) mdot = mdot + dmdot
else
  prod = delpt(j)*delpt(j-1)
  if (delpt(j) .lt. 0. .and. prod .gt. 0.) mdot = mdot - dmdot
  if (delpt(j) .gt. 0. .and. prod .gt. 0.) mdot = mdot + dmdot
  if (delpt(j) .lt. 0. .and. prod .lt. 0.) then
    dmdot = dmdot/2.
    mdot = mdot - dmdot
  endif
  if (delpt(j) .gt. 0. .and. prod .lt. 0.) then
    dmdot = dmdot/2.
    mdot = mdot + dmdot
  endif
endif
10 continue

```

```

if (abs(dpt) .gt. tol) write (6,1002) time,dpt
1002 format ('*** tank pressure does not converge at time = ',f10.2,
1      ' sec,', ' delpt = ',e9.3,' psi ***')
15 if (time .ge. prtsp .and. time .le. (prtsp + 0.1)) then
dppump = (psmi - pt)/144.
dpsm = dpsm/144.
dpsi = (pi - pt)/144.
hliq = 12.*zliq
write (6,1)
1 format (/5x,'SPRAY MANIFOLD/INJECTION TUBE FLOW MODEL')
write (6,2) acc,pull,pman,hliq,rho,visc
2 format (5x,'Acceleration Level      ',f6.1,' g')
1   5x,'Ullage Pressure          ',f6.3,' psia'
2   5x,'Spray Manifold Inlet Pressure  ',f6.3,' psia'
3   5x,'Liquid Level            ',f6.1,' in'
4   5x,'Liquid Density          ',f6.3,' lbm/ft3'
5   5x,'Liquid Viscosity         ',e9.3,' lbm/ft-sec')
write (6,3) mdot,dppump,dpsm,dpsi
3 format (5x,'Pump Flow Rate      ',f6.3,' lbm/sec')
1   5x,'Pump Pressure Rise     ',f6.3,' psi'
2   5x,'Spray Manifold Tube Delta p  ',f6.3,' psi'
3   5x,'Spray Injection Tube Delta p  ',f6.3,' psi')
write (6,4)
4 format (5x,'Node',2x,'Distance',3x,'Inlet',2x,'Outlet',
1   3x,'Nodal',3x,'Tank',5x,'Inlet',4x,'Outlet',
2   3x,'Injection',3x,'Injection',3x,'Orifice')
write (6,5)
5 format (24x,'p',7x,'p',7x,'p',7x,'p',
1   7x,'mdot',5x,'mdot',7x,'mdot',6x,'Velocity',
2   5x,'CdA')
write (6,6)
6 format (13x,'(in)',4x,'(psia)',2x,'(psia)',2x,'(psia)',2x,'(psia)',
1   2x,'(lbm/sec)',1x,'(lbm/sec)',1x,'(lbm/sec)',5x,'(fps)',
2   6x,'(in2)')
do i = 1,n
    pin(i) = pin(i)/144.
    pout(i) = pout(i)/144.
    pnode(i) = pnode(i)/144.
    ptank(i) = ptank(i)/144.
    asp(i) = 144.*as(i)
    x(i) = 12.*x(i)
    write (6,7) i,x(i),pin(i),pout(i),pnode(i),ptank(i),
1      mdin(i),mdout(i),mds(i),veld(i),asp(i)
7 format (5x,i3,4x,f6.2,3x,4(f6.3,2x),
1      3(e9.3,1x),4x,f5.2,4x,f8.6)

```

```

enddo
endif
return
end

subroutine value(np,x,y,xin,yout)
c
c This subroutine performs lagrangian interpolation within a set
c of (x,y) pairs to give yout corresponding to xin
c
dimension x(np),y(np)
if (xin .le. x(1)) yout = y(1)
if (xin .le. x(1)) return
if (xin .ge. x(np)) yout = y(np)
if (xin .ge. x(np)) return
do 10 i = 1,np
    k = i
    if (xin .lt. x(i)) go to 30
10 continue
30 ffr = (xin - x(k - 1))/(x(k) - x(k - 1))
    yout = y(k - 1) + ffr*(y(k) - y(k - 1))
    return
end

```

SUBROUTINE VALUE2(NPX,NPY,X,Y,Z,XIN,YIN,ZOUT)
 DIMENSION X(NPX),Y(NPY),Z(NPX,NPY)

```

C
IF(XIN .LE. X(1) .AND. YIN .LE. Y(1))ZOUT=Z(1,1)
IF(XIN .LE. X(1) .AND. YIN .LE. Y(1))RETURN
C
IF(XIN .GE. X(NPX) .AND. YIN .LE. Y(1))ZOUT=Z(NPX,1)
IF(XIN .GE. X(NPX) .AND. YIN .LE. Y(1))RETURN
C
IF(XIN .LE. X(1) .AND. YIN .GE. Y(NPY))ZOUT=Z(1,NPY)
IF(XIN .LE. X(1) .AND. YIN .GE. Y(NPY))RETURN
C
IF(XIN .GE. X(NPX) .AND. YIN .GE. Y(NPY))ZOUT=Z(NPX,NPY)
IF(XIN .GE. X(NPX) .AND. YIN .GE. Y(NPY))RETURN
C
IF(XIN .GT. X(1))GO TO 30
DO 20 I=1,NPY
    M=I
    IF(YIN .LT. Y(I))GO TO 25
20 CONTINUE
25 FFRY=(YIN-Y(M-1))/(Y(M)-Y(M-1))

```

```

ZOUT=Z(1,M-1)+FFRY*(Z(1,M)-Z(1,M-1))
RETURN
C
30 IF(XIN .LT. X(NPX))GO TO 60
DO 50 I=1,NPY
    M=I
    IF(YIN .LT. Y(I))GO TO 55
50 CONTINUE
55 FFRY=(YIN-Y(M-1))/(Y(M)-Y(M-1))
    ZOUT=Z(NPX,M-1)+FFRY*(Z(NPX,M)-Z(NPX,M-1))
    RETURN
C
60 IF(YIN .GT. Y(1))GO TO 90
DO 80 I=1,NPX
    L=I
    IF(XIN .LT. X(I))GO TO 85
80 CONTINUE
85 FFRX=(XIN-X(L-1))/(X(L)-X(L-1))
    ZOUT=Z(L-1,1)+FFRX*(Z(L,1)-Z(L-1,1))
    RETURN
C
90 IF(YIN .LT. Y(NPY))GO TO 120
DO 110 I=1,NPX
    L=I
    IF(XIN .LT. X(I))GO TO 115
110 CONTINUE
115 FFRX=(XIN-X(L-1))/(X(L)-X(L-1))
    ZOUT=Z(L-1,NPY)+FFRX*(Z(L,NPY)-Z(L-1,NPY))
    RETURN
C
120 DO 130 I=1,NPX
    L=I
    IF(XIN .LT. X(I))GO TO 135
130 CONTINUE
135 FXR=(XIN-X(L-1))/(X(L)-X(L-1))
    DO 140 I=1,NPY
        M=I
        IF(YIN .LT. Y(I))GO TO 145
140 CONTINUE
145 FYR=(YIN-Y(M-1))/(Y(M)-Y(M-1))
C
    ZXLO=Z(L-1,M-1)+FYR*(Z(L-1,M)-Z(L-1,M-1))
    ZXHI=Z(L,M-1)+FYR*(Z(L,M)-Z(L,M-1))
C
    ZOUT=ZXLO+FXR*(ZXHI-ZXLO)

```

RETURN

END

subroutine volarea(vt)

c

c subroutine volarea.f to calculate the volumes and areas of
c an elliptical bulkhead tank

c

real l

common /tankdim/do,l,ho,t,c1
common /tankvol/v1,v2,vbi,vbo,vci,vco
common /tankarea/abi,aci,ati
ro = do/2.
ri = ro - t
hi = ho - t
xk1 = ri/hi
c1 = xk1**4 - xk1*xk1

c

c internal bulkhead

c

vbi = 2./3.*3.14159*ri*ri*hi
abi = 3.14159*(hi*(c1*hi*hi + ri*ri)**.5
1 + ri*ri/c1**.5*(log(hi*c1**.5
2 + (c1*hi*hi + ri*ri)**.5) - log(ri)))

c

c external bulkhead

c

vbo = 2./3.*3.14159*ro*ro*ho
abo = 3.14159*(ho*(c1*ho*ho + ro*ro)**.5
1 + ro*ro/c1**.5*(log(ho*c1**.5
2 + (c1*ho*ho + ro*ro)**.5) - log(ro)))

c

c internal cylinder

c

vci = 3.14159*ri*ri*l
aci = 2.*3.14159*ri*l

c

c external cylinder

c

vco = 3.14159*ro*ro*l
aco = 2.*3.14159*ro*l

c

c tank

c

v1 = vbi

```

v2 = vbi + vci
vt = 2.*vbi + vci
ati = 2.*abi + aci
return
end

subroutine vwall(hliq,vtw)
c
c subroutine vwall.f to calculate the wall volume exposed to
c the ullage gas
c
real 1
common /tankdim/do,l,ho,t,c1
common /tankvol/v1,v2,vbi,vbo,vci,vco
ro = do/2.
ri = ro - t
hi = ho - t
vbw = vbo - vbi
vcw = vco - vci
if (hliq .gt. (hi + l)) then
    h = hliq - (hi + l)
    vi = 3.14159*h*(ri/hi)**2*(hi*hi - h*h/3.)
    vo = 3.14159*h*(ro/ho)**2*(ho*ho - h*h/3.)
    v = vo - vi
    vtw = vbw - v
else
    if (hliq .gt. hi) then
        h = hi + l - hliq
        vi = 3.14159*ri*ri*h
        vo = 3.14159*ro*ro*h
        v = vo - vi
        vtw = vbw + v
    else
        h = hi - hliq
        vi = 3.14159*h*(ri/hi)**2*(hi*hi - h*h/3.)
        vo = 3.14159*h*(ro/ho)**2*(ho*ho - h*h/3.)
        v = vo - vi
        vtw = vbw + vcw + v
    endif
endif
return
end

```

3.5 Input Data

3.5.1 Heat Exchanger Model

```
*****|*****|*****|*****|*****|*****|  
MASSIC | PIC | PSTI |MDOT(VNT) | PBACK | Program is vent2  
*****|*****|*****|*****|*****|*****|  
270.0 20.590 20.645 0.00475 19.5  
*****|*****|*****|*****|*****|*****|  
PROP | PTP | AVLIQ | PAMB | G |  
*****|*****|*****|*****|*****|*****|  
1.0 1.021 3500.0 0.0 0.0  
*****|*****|*****|*****|*****|*****|  
M | VT | AX | VA | Twall |  
*****|*****|*****|*****|*****|*****|  
25 68.9342 0.04500 0 32.00  
*****|*****|*****|*****|*****|*****|  
MDOT(sp)| DI (sp) | THKNESS | VFLW DI | E/D |  
*****|*****|*****|*****|*****|*****|  
0.32 1.18 0.035 0.25 1.0e-6  
*****|*****|*****|*****|*****|*****|  
DPINC | ERRMX | PERRMX | DEBUG | EQ DIAM |  
*****|*****|*****|*****|*****|*****|  
0.005 0.005 0.0010 0.0 0.134  
*****|*****|*****|*****|*****|*****|  
FINTIM | PRDEL | QDTERR | DELT | DELPTP |  
*****|*****|*****|*****|*****|*****|  
2.2 0.02 0.03 0.01 0.08  
*****|*****|*****|*****|*****|*****|  
OT | OM | OP | DEL Qdotl |  
*****|*****|*****|*****|*****|*****|  
0 0 0 0.0010  
*****|*****|*****|*****|*****|  
A(1-M) | DH(1-M) | QDOT(1-M) | LENGTH |  
*****|*****|*****|*****|*****|  
0.04500 0.0 1.00 0.0  
0.27721 0.0 1.00 6.0  
0.27721 0.0 0.975 6.0  
0.27721 0.0 0.95 6.0  
0.27721 0.0 0.925 6.0  
0.27721 0.0 0.90 6.0  
0.27721 0.0 0.875 6.0  
0.27721 0.0 0.85 6.0  
0.27721 0.0 0.825 6.0  
0.27721 0.0 0.80 6.0
```

0.27721	0.0	0.75	6.0
0.27721	0.0	0.70	6.0
0.27721	0.0	0.65	6.0
0.27721	0.0	0.60	6.0
0.27721	0.0	0.500	6.0
0.27721	0.0	0.450	4.0
0.27721	0.0	0.400	4.0
0.27721	0.0	0.350	4.0
0.27721	0.0	0.300	4.0
0.27721	0.0	0.250	4.0
0.27721	0.0	0.200	4.0
0.27721	0.0	0.150	4.0
0.27721	0.0	0.100	4.0
0.27721	0.0	0.050	4.0
0.00372	0.0	0.000	0.0

Simulation of LH2 Vent Thru Zero g Vent System Heat Exchanger (4/6/93)
 Baseline Vent Area of 0.00372 in² (Mdot & Qdot Trade) delta q = 0.0010

H2 MISC DATA

LMXPAR

18	
-2.0	2.107
-1.699	1.835
-1.3979	1.585
-1.1549	1.3874
-1.0	1.2672
-0.6990	1.0492
-0.3979	0.8482
-0.1549	0.7024
0.0	0.6232
0.3010	0.4914
0.6020	0.3766
0.8451	0.2923
1.0	0.2430
1.301	0.1703
1.602	0.1106
1.8451	0.0682
2.0	0.0453
10.0	0.04

LVISP

31	
0.0	1.985e-5
1.021	1.751e-5

2.553	1.4e-5
4.17	1.24e-5
6.446	1.11e-5
9.527	1.006e-5
13.561	0.917e-5
15.984	0.877e-5
18.694	0.84e-5
21.723	0.805e-5
25.089	0.773e-5
28.813	0.742e-5
32.915	0.713e-5
37.415	0.685e-5
42.334	0.659e-5
47.693	0.634e-5
53.514	0.609e-5
59.817	0.586e-5
66.625	0.563e-5
73.957	0.54e-5
81.838	0.519e-5
90.287	0.497e-5
99.329	0.476e-5
108.987	0.454e-5
119.297	0.433e-5
130.299	0.410e-5
142.027	0.387e-5
154.522	0.362e-5
167.848	0.333e-5
182.136	0.292e-5
187.510	0.24e-5

GVISP

24

0.0	0.045e-5
1.021	0.05e-5
2.553	0.057e-5
9.527	0.07e-5
18.694	0.079e-5
32.915	0.089e-5
37.415	0.092e-5
42.334	0.095e-5
47.693	0.097e-5
53.514	0.1e-5
59.817	0.103e-5
66.625	0.106e-5
73.957	0.109e-5

81.838	0.112e-5
90.287	0.116e-5
99.329	0.12e-5
108.987	0.124e-5
119.297	0.129e-5
130.299	0.135e-5
142.027	0.143e-5
154.522	0.153e-5
167.848	0.168e-5
182.136	0.196e-5
187.51	0.24e-5

H2 RHO DATA

LRHOP

39	
0.0325	5.385
1.01	5.385
1.021	4.80827
1.073	4.80377
1.462	4.77434
1.950	4.74423
2.553	4.71359
3.288	4.68219
4.170	4.65003
5.217	4.61705
6.446	4.58321
7.877	4.54844
9.527	4.51266
11.416	4.47582
13.561	4.43782
15.984	4.39858
18.694	4.35801
21.723	4.31600
25.089	4.27243
28.813	4.22718
32.915	4.18010
37.415	4.13102
42.334	4.07975
47.693	4.02608
53.514	3.96975
59.817	3.9105
66.625	3.8479
73.957	3.7815
81.838	3.7108
90.287	3.635

99.33	3.5534
108.987	3.4646
119.297	3.3668
130.299	3.2572
142.027	3.1314
154.522	2.9808
167.848	2.7851
182.136	2.4567
187.51	1.9620

GRHOP

40

0.0325	0.000345
0.1137	0.00108
0.3163	0.002742
1.021	0.00784
1.073	0.00819
1.462	0.01077
1.950	0.01391
2.553	0.01765
3.288	0.02207
4.170	0.02724
5.217	0.03322
6.446	0.04008
7.877	0.04790
9.527	0.05675
11.416	0.06671
13.561	0.07787
15.984	0.09011
18.694	0.10395
21.723	0.11930
25.089	0.13629
28.813	0.15504
32.915	0.17568
37.415	0.19839
42.334	0.22335
47.693	0.25078
53.514	0.28094
59.817	0.3141
66.625	0.35076
73.957	0.39126
81.838	0.43621
90.287	0.48635
99.329	0.54266
108.987	0.60647

119.297	0.67967
130.299	0.76511
142.027	0.86751
154.522	0.99566
167.848	1.17023
182.136	1.4779
187.510	1.96202

RHOVP

20

0.000345	0.0325
0.00108	0.1137
0.002742	0.3163
0.00784	1.021
0.00819	1.073
0.01077	1.462
0.01391	1.950
0.01765	2.553
0.02207	3.288
0.02724	4.170
0.03322	5.217
0.04008	6.446
0.0479	7.877
0.05675	9.527
0.06671	11.416
0.07787	13.561
0.09011	15.984
0.10395	18.694
0.11930	21.723
0.13629	25.089

SCLENT

10 7

10.0 3

34.0	-115.878
34.263	-115.318
36.483	-110.59

14.7 4

34.0	-115.740
36.0	-111.351
36.483	-110.240
36.608	-109.95

15.0 4

34.0	-115.731
36.0	-111.343

36.608	-109.942
38.444	-105.71
20.0	5
34.0	-115.583
36.0	-111.201
38.0	-106.489
38.444	-105.395
39.975	-101.62
25.0	5
34.0	-115.435
36.0	-111.059
38.0	-106.356
39.975	-101.358
41.299	-98.01
30.0	6
34.0	-115.287
36.0	-110.917
38.0	-106.223
40.0	-101.170
41.299	-97.667
42.475	-94.5
35.0	7
34.0	-115.139
36.0	-110.775
38.0	-106.090
40.0	-101.052
42.0	-95.584
42.475	-94.227
43.536	-91.2
40.0	7
34.0	-114.991
36.0	-110.633
38.0	-105.957
40.0	-100.929
42.0	-95.477
43.536	-90.977
45.406	-85.50
45.0	7
34.0	-114.842
36.0	-110.491
38.0	-105.823
40.0	-100.806
42.0	-95.370
44.0	-89.454
45.406	-85.08

50.0	7
34.0	-114.694
36.0	-110.348
38.0	-105.689
40.0	-100.682
42.0	-95.262
44.0	-89.366
45.406	-84.890

LH2 SATURATED PROPERTIES

* Pressure	* Cp	* Thermal Cond	* Surface T	* Sat Temp	*
16					
1.021	1.557	0.04199	1.7076e-5	24.845	
1.462	1.619	0.04551	1.6469e-5	26.0	
2.553	1.723	0.05004	1.5419e-5	28.0	
4.17	1.849	0.05297	1.4374e-5	30.0	
6.446	1.985	0.05489	1.3333e-5	32.0	
9.527	2.125	0.05601	1.2298e-5	34.0	
13.561	2.27	0.05690	1.1267e-5	36.0	
18.694	2.443	0.05793	1.0243e-5	38.0	
25.089	2.637	0.05843	0.9225e-5	40.0	
28.813	2.743	0.05851	0.8718e-5	41.0	
32.915	2.848	0.05848	0.8213e-5	42.0	
42.334	3.097	0.05811	0.7209e-5	44.0	
53.514	3.393	0.05734	0.6214e-5	46.0	
66.625	3.772	0.05616	0.5228e-5	48.0	
81.838	4.307	0.05459	0.4253e-5	50.0	
99.329	5.074	0.05258	0.3292e-5	52.0	

GH2 PROPERTIES

* Cp	* Thermal Cond	* Viscosity	*
6 14	50.0		
1.0			
2.511	7.21e-3	5.0e-7	
2.506	7.35e-3	5.2e-7	
2.499	7.63e-3	5.6e-7	
2.493	7.92e-3	6.1e-7	
2.489	8.24e-3	6.5e-7	
2.486	8.56e-3	6.9e-7	
2.483	8.96e-3	7.3e-7	
2.481	9.47e-3	7.7e-7	
2.479	9.98e-3	8.1e-7	
2.478	1.049e-2	8.5e-7	
2.477	1.10e-2	8.9e-7	
2.475	1.151e-2	9.3e-7	
2.474	1.202e-2	9.7e-7	

2.474	1.252e-2	1.0e-6
5.0		
2.652	6.33e-3	6.3e-7
2.636	7.815e-3	6.43e-7
2.610	8.707e-3	6.72e-7
2.589	8.879e-3	7.03e-7
2.574	9.174e-3	7.35e-7
2.561	9.550e-3	7.73e-7
2.551	9.933e-3	8.05e-7
2.543	1.032e-2	8.32e-7
2.536	1.070e-2	8.56e-7
2.524	1.109e-2	8.87e-7
2.524	1.148e-2	9.18e-7
2.52	1.186e-2	9.49e-7
2.516	1.224e-2	9.79e-7
2.512	1.262e-2	1.01e-6
10.0		
2.790	7.04e-3	7.1e-7
2.768	7.892e-3	7.21e-7
2.726	9.425e-3	7.43e-7
2.696	9.714e-3	7.68e-7
2.671	1.001e-2	7.93e-7
2.649	1.030e-2	8.19e-7
2.633	1.061e-2	8.38e-7
2.619	1.092e-2	8.61e-7
2.607	1.122e-2	8.93e-7
2.596	1.153e-2	9.19e-7
2.587	1.183e-2	9.43e-7
2.578	1.214e-2	9.62e-7
2.571	1.245e-2	9.85e-7
2.564	1.276e-2	1.01e-6
15.0		
2.912	9.84e-3	7.6e-7
2.886	9.95e-3	7.72e-7
2.837	1.017e-2	7.94e-7
2.800	1.040e-2	8.16e-7
2.766	1.064e-2	8.37e-7
2.740	1.088e-2	8.59e-7
2.717	1.113e-2	8.78e-7
2.698	1.138e-2	8.94e-7
2.681	1.163e-2	9.13e-7
2.666	1.188e-2	9.34e-7
2.653	1.213e-2	9.56e-7
2.641	1.239e-2	9.77e-7
2.631	1.264e-2	9.99e-7

2.621	1.290e-2	1.02e-6
20.0		
3.029	1.054e-2	8.0e-7
3.001	1.063e-2	8.09e-7
2.946	1.080e-2	8.27e-7
2.904	1.098e-2	8.45e-7
2.865	1.117e-2	8.64e-7
2.833	1.137e-2	8.82e-7
2.805	1.158e-2	9.01e-7
2.781	1.178e-2	9.19e-7
2.760	1.199e-2	9.38e-7
2.741	1.220e-2	9.56e-7
2.725	1.241e-2	9.75e-7
2.710	1.262e-2	9.92e-7
2.697	1.284e-2	1.01e-6
2.684	1.305e-2	1.02e-6
30.0		
3.261	1.173e-2	8.7e-7
3.228	1.178e-2	8.8e-7
3.171	1.189e-2	8.95e-7
3.121	1.201e-2	9.06e-7
3.072	1.212e-2	9.16e-7
3.032	1.225e-2	9.29e-7
2.998	1.239e-2	9.43e-7
2.964	1.253e-2	9.57e-7
2.937	1.267e-2	9.70e-7
2.912	1.281e-2	9.84e-7
2.887	1.295e-2	9.98e-7
2.868	1.310e-2	1.01e-6
2.849	1.324e-2	1.02e-6
2.83	1.339e-2	1.02e-6
0.0		
0.04		
0.12		
0.2		
0.28		
0.36		
0.44		
0.52		
0.6		
0.68		
0.76		
0.84		
0.92		
1.0		

```

*****
pu zliq ztank dmdot
*****
20. 120. 120. .005
*****
zsm dsi zsi n nbar
*****
132. .444 120. 45 4
*****
ks acc rho visc roverd
*****
.5 1. 4.339 0.817e-5 2.
*****
tol nlim
*****
.001 50
*****
opn opt omdin omds ovel ocdas
*****
0. 0. 0. 0. 0. 0.
20 PSIA ULLAGE, 0.2 LBM/SEC, 1 G, FULL TANK, 45 ORF
DISTANCE FROM SPRAY TUBE INLET (IN)
PRESSURE (PSIA)
FLOW RATE THROUGH SPRAY TUBE (LBM/SEC)
INJECTION FLOW RATE (LBM/SEC)
INJECTION VELOCITY (FT/SEC)
ORIFICE CDA (IN2)
8
4 .001901
8 .002224
12 .002534
16 .002821
20 .003079
25 .003629
30 .004122
45 .004647

```

3.5.2 Integrated Zero-g TVS Model

```

***** tvs.f
xd xchar he xcond prtsp outp           input
***** data
1.0 0.25 0.0 1.0 13200.0 1.0
*****
```

```

pui    tui    pli    twi    twli    full    xl1
*****
20.0   38.431  20.0   38.841  38.841  10.0    5.375
*****
dtank   hcyl   hbulk   tkw    dsb    d1    d2
*****
120.0   60.0   30.0   0.5    0.25   1.902   1.402
*****
mdvent  mdsi   dthex   qflux   g     hwliq
*****
0.0052  0.001  2.2167  0.25   1.0e-6  25.0
*****
pmin    pmax   delt2   iprnt2  iplot2
*****
20.0   21.0   1.0    15     15
*****
fintim  delt1   xdelt1  iprnt1  iplot1  nline
*****
12500.0 0.1    0.1    10     10     40
*****
opu    otu    omu    opl    otl    oml
*****
1.0    0.0    0.0    1.0    0.0    0.0
*****
otw    omwl   omdv   omdu   omdbw
*****
0.0    0.0    1.0    0.0    0.0    0.0
*****
omdlu  omdul   omds   onpump  odppmp
*****
0.0    0.0    1.0    1.0    1.0

```

TIME (SEC)

ULLAGE PRESSURE (PSIA)

ULLAGE TEMPERATURE (R)

ULLAGE MASS (LBM)

BULK LIQUID PRESSURE (PSIA)

BULK LIQUID TEMPERATURE (R)

BULK LIQUID MASS (LBM)

WALL TEMPERATURE (ULLAGE SIDE) (R)

WALL LIQUID MASS (LBM)

OVERBOARD VENT FLOW RATE (LBM/SEC)

ULLAGE SPRAY FLOW RATE (LBM/SEC)

LIQUID DROPLET BOILING RATE (LBM/SEC)

WALL LIQUID BOILING RATE (LBM/SEC)

INTERFACIAL MASS-TRANSFER RATE (LBM/SEC)

PUMP FLOW RATE (LBM/SEC)

PUMP SPEED (RPM)

PUMP PRESSURE RISE (PSI)

ZERO-g TVS TRANSIENT PERFORMANCE

(0g, 10% FULL, 0.25 BTU/HR-FT2, NO He)

***** pump.f

mdotd dpd npumpi npumpd xhp xn input

***** data

0.3 0.500 0.0 3134.0 5.0 0.01

deltat effp

0.2 0.65

***** spray.f

dsm zsm dsi zsi norf nbar input

***** data

1.18 132.0 0.444 120.0 45 4

ks cds roverd dmdot tol nlim

1.56 0.8 2.0 0.005 0.001 200

1

45 0.0670

24 p h cv cp rho beta k mu Saturated hydrogen
properties

24.845	1.021	-132.892	1.126	1.557	4.80827	.0058609	.04199	1.751e-5
		60.357	1.484	2.513	.00784	.0417855	.00719	.050e-5
25.	1.073	-132.647	1.129	1.568	4.80377	.0059353	.04250	1.730e-5
		60.699	1.484	2.518	.00819	.0415883	.00721	.050e-5
26.	1.462	-131.030	1.151	1.619	4.77434	.0061184	.04551	1.605e-5
		62.879	1.487	2.532	.01077	.0404077	.00739	.052e-5
27.	1.950	-129.360	1.174	1.669	4.74429	.0063005	.04800	1.496e-5
		65.002	1.491	2.551	.01391	.0393821	.00756	.054e-5
28.	2.553	-127.633	1.198	1.723	4.71359	.0064909	.05004	1.400e-5
		67.062	1.496	2.573	.01765	.0384993	.00775	.057e-5
29.	3.288	-125.846	1.221	1.786	4.68219	.0067848	.05168	1.315e-5
		69.056	1.501	2.598	.02207	.0377489	.00793	.059e-5
30.	4.170	-123.995	1.245	1.849	4.65003	.0070405	.05297	1.240e-5
		70.977	1.507	2.627	.02724	.0371221	.00813	.061e-5
31.	5.217	-122.077	1.267	1.915	4.61705	.0073258	.05402	1.172e-5
		72.821	1.513	2.659	.03322	.0366118	.00834	.063e-5

32.	6.446	-120.090	1.289	1.985	4.58321	.0076421	.05489	1.112e-5
		74.584	1.520	2.695	.04008	.0362124	.00856	.066e-5
33.	7.877	-118.029	1.310	2.051	4.54844	.0079185	.05555	1.056e-5
		76.261	1.527	2.734	.04790	.0359198	.00880	.068e-5
34.	9.527	-115.893	1.329	2.125	4.51266	.0082741	.05601	1.006e-5
		77.848	1.535	2.778	.05675	.0357318	.00904	.070e-5
35.	11.416	-113.678	1.348	2.195	4.47582	.0085840	.05631	.959e-5
		79.339	1.543	2.826	.06671	.0356477	.00929	.072e-5
36.	13.561	-111.380	1.365	2.270	4.43782	.0089505	.05690	.917e-5
		80.729	1.551	2.879	.07787	.0356683	.00962	.075e-5
37.	15.984	-108.997	1.380	2.360	4.39858	.0094522	.05748	.877e-5
		82.105	1.559	2.935	.09011	.0357690	.00998	.077e-5
38.	18.694	-106.524	1.395	2.443	4.35801	.0099007	.05793	.840e-5
		83.256	1.567	2.998	.10395	.0360086	.01036	.079e-5
39.	21.723	-103.956	1.408	2.532	4.31600	.0103939	.05824	.805e-5
		84.290	1.576	3.068	.11930	.0363705	.01076	.082e-5
40.	25.089	-101.289	1.420	2.637	4.27243	.0110382	.05843	.773e-5
		85.199	1.584	3.146	.13629	.0368634	.01117	.084e-5
41.	28.813	-98.517	1.431	2.743	4.22718	.0117045	.05851	.742e-5
		85.976	1.592	3.233	.15504	.0374994	.01159	.087e-5
42.	32.915	-95.636	1.441	2.848	4.18010	.0123677	.05848	.713e-5
		86.614	1.601	3.331	.17568	.0382948	.01204	.089e-5
43.	37.415	-92.637	1.451	2.969	4.13102	.0131689	.05835	.685e-5
		87.103	1.610	3.441	.19839	.0392710	.01251	.092e-5
44.	42.334	-89.513	1.459	3.097	4.07975	.0140457	.05811	.659e-5
		87.431	1.619	3.566	.22335	.0404560	.01301	.095e-5
45.	47.693	-86.254	1.467	3.242	4.02608	.0150803	.05778	.634e-5
		87.584	1.629	3.709	.25078	.0418864	.01354	.097e-5
50.	81.838	-67.493	1.507	4.307	3.71079	.0235331	.05459	.519e-5
		85.043	1.693	4.919	.43621	.0551860	.01694	.112e-5
55.	130.299	-42.122	1.564	7.528	3.25720	.0531428	.04960	.410e-5
		73.413	1.831	9.187	.76511	.1048799	.02422	.135e-5
8 11 h	cv	cp	rho	beta	k	mu	T	Superheated hydrogen properties
0.	.01	.02	.03	.04	.05	.06	.07	.08
1.	.09	.1	600.					
1.	25.000							
60.765	1.483	2.511	.00763	.00721	.050e-5	25.000		
75.138	1.481	2.492	.00617	.00804	.063e-5	30.750		
89.438	1.480	2.483	.00518	.00915	.074e-5	36.500		
103.700	1.480	2.478	.00447	.01055	.086e-5	42.250		
117.938	1.480	2.474	.00393	.01202	.097e-5	48.000		
132.160	1.480	2.473	.00351	.01347	.108e-5	53.750		
146.372	1.480	2.471	.00317	.01477	.117e-5	59.500		
160.577	1.480	2.470	.00297	.01607	.128e-5	65.250		
174.782	1.481	2.470	.00265	.01737	.137e-5	71.000		
187.139	1.484	2.472	.00248	.01850	.146e-5	76.750		

203.223	1.489	2.477	.00228	.01996	.156e-5	82.500
10.	30.806					
78.249	1.537	2.790	.05926	.00704	.071e-5	34.263
93.566	1.502	2.649	.04952	.01030	.082e-5	39.920
108.372	1.493	2.591	.04274	.01168	.093e-5	45.578
122.930	1.489	2.558	.03766	.01306	.103e-5	51.235
137.339	1.487	2.537	.03370	.01439	.114e-5	56.892
151.651	1.486	2.523	.03051	.01564	.124e-5	62.550
165.892	1.485	2.513	.02788	.01692	.133e-5	68.207
180.088	1.486	2.506	.02570	.01819	.142e-5	73.865
194.258	1.489	2.503	.02382	.01944	.151e-5	79.522
208.423	1.493	2.504	.02220	.02071	.160e-5	85.179
222.605	1.502	2.509	.02080	.02198	.169e-5	90.837
15.	36.608					
81.613	1.556	2.912	.08508	.00984	.076e-5	36.608
97.298	1.511	2.722	.07143	.01107	.087e-5	42.242
112.381	1.499	2.640	.06178	.01239	.098e-5	47.876
127.120	1.493	2.595	.05459	.01374	.108e-5	53.510
141.657	1.490	2.566	.04898	.01500	.118e-5	59.144
156.056	1.489	2.546	.04445	.01625	.128e-5	64.778
170.360	1.488	2.532	.04072	.01751	.137e-5	70.412
184.599	1.489	2.524	.03759	.01876	.147e-5	76.045
198.803	1.492	2.519	.03491	.02001	.155e-5	81.679
212.996	1.498	2.519	.03260	.02126	.164e-5	87.313
227.201	1.507	2.524	.03057	.02252	.172e-5	92.947
20	38.444					
83.731	1.571	3.029	.11058	.01054	.080e-5	38.444
99.896	1.519	2.790	.09269	.01169	.091e-5	44.060
115.240	1.504	2.689	.08029	.01297	.102e-5	49.675
130.163	1.498	2.631	.07106	.01427	.112e-5	55.291
144.826	1.494	2.594	.06384	.01550	.121e-5	60.906
159.318	1.491	2.568	.05801	.01674	.131e-5	66.522
173.683	1.490	2.552	.05327	.01799	.140e-5	72.137
187.979	1.492	2.541	.04921	.01923	.149e-5	77.753
202.224	1.495	2.534	.04574	.02047	.158e-5	83.368
216.452	1.502	2.533	.04273	.02171	.167e-5	88.984
230.687	1.512	2.538	.04010	.02296	.175e-5	94.600
25	39.975					
85.177	1.584	3.144	.13584	.01116	.084e-5	39.975
101.806	1.526	2.857	.11368	.01224	.094e-5	45.575
117.424	1.509	2.734	.09846	.01348	.105e-5	51.176
132.526	1.501	2.665	.08719	.01473	.115e-5	56.776
147.319	1.497	2.620	.07839	.01594	.125e-5	62.376
161.903	1.493	2.590	.07131	.01717	.134e-5	67.976
176.339	1.493	2.570	.06554	.01841	.143e-5	73.577

190.690	1.494	2.556	.06058	.01964	.152e-5	79.177
204.980	1.498	2.548	.05634	.02087	.161e-5	84.177
219.242	1.505	2.546	.05269	.02210	.169e-5	90.377
233.514	1.517	2.550	.04952	.02335	.177e-5	95.978
30.	41.299					
86.182	1.595	3.261	.16101	.01173	.087e-5	41.299
103.265	1.533	2.924	.13441	.01274	.098e-5	46.886
119.156	1.514	2.779	.11635	.01393	.108e-5	52.473
134.439	1.505	2.697	.10303	.01523	.117e-5	58.060
149.354	1.499	2.646	.09273	.01634	.127e-5	63.647
164.030	1.495	2.610	.08442	.01756	.136e-5	69.234
178.543	1.495	2.587	.07756	.01878	.146e-5	74.821
192.947	1.496	2.571	.07178	.02000	.155e-5	80.408
207.282	1.500	2.562	.06685	.02122	.163e-5	85.995
221.586	1.509	2.559	.06257	.02245	.171e-5	91.582
235.893	1.521	2.563	.05881	.02369	.179e-5	97.169
35.	42.475					
86.865	1.605	3.381	.18169	.01226	.090e-5	42.475
104.432	1.539	2.987	.15482	.01319	.100e-5	48.050
120.569	1.518	2.822	.13404	.01434	.110e-5	53.626
136.023	1.508	2.730	.11876	.01551	.120e-5	59.201
151.065	1.501	2.670	.10690	.01671	.129e-5	64.776
165.833	1.498	2.630	.09737	.01791	.139e-5	70.351
180.418	1.497	2.604	.08953	.01912	.147e-5	75.927
194.879	1.499	2.586	.08291	.02033	.156e-5	81.502
209.264	1.503	2.576	.07724	.02155	.165e-5	87.077
223.610	1.512	2.572	.07231	.02277	.173e-5	92.652
237.956	1.524	2.575	.06799	.02399	.181e-5	98.228
40.	43.536					
87.300	1.615	3.506	.21148	.01278	.093e-5	43.536
105.311	1.545	3.056	.17544	.01362	.103e-5	49.101
121.728	1.523	2.866	.15163	.01472	.113e-5	54.665
137.369	1.512	2.761	.13427	.01587	.122e-5	60.230
152.534	1.503	2.694	.12088	.01705	.132e-5	65.795
167.379	1.500	2.651	.11027	.01823	.140e-5	71.359
182.041	1.499	2.621	.10139	.01943	.149e-5	76.924
196.562	1.500	2.601	.09390	.02063	.158e-5	82.488
210.999	1.506	2.589	.08750	.02185	.166e-5	88.053
225.389	1.515	2.584	.08193	.02306	.175e-5	93.618
239.769	1.528	2.586	.07706	.02428	.183e-5	99.182